OWNER'S HANDBOOK I

OWNER'S HANDBOOK I

WORKING WITH CORNERSTONE

CORNERSTONE™

INFOCOM™

OWNER'S HANDBOOK I

INFOCOM

Cornerstone designed and built by:	Brian Berkowitz (principal designer) Richard Ilson (principal designer) Paul DiLascia Craig Leckband Ed Black (project manager) Chris Reeve (consultant) Tim Anderson (consultant)
Cornerstone documentation:	Richard Ilson (editorial manager) Elizabeth Young (production manager)
Written by:	Paul DiLascia Barry Jacobson Larry Martin Mike Quinn Mark Sawtelle
Graphic design:	Marsha Caine
Typography:	Woodland Graphics, Bedford, MA
Printed and bound by:	CSA Press, Hudson, MA
Illustrations:	Jim Carson © 1984
Diagrams:	Suzanne L. Breault © 1984

Printed in U.S.A.

Infocom, Inc.
55 Wheeler St.
Cambridge, MA 02138

First printing January 1985.

OWNER'S HANDBOOK I

TABLE OF CONTENTS

Preface

This *Owner's Handbook* comes in two volumes. Volume I, "Working With Cornerstone," covers the major uses of the system. Volume II, "Additional Information and Advanced Concepts," contains information less central to daily use (such as how to use Cornerstone's conversion and backup utilities).

The examples in Volume I frequently refer to the Sample Database included as one of your application diskettes. The Sample Database serves as an order entry system at Omnifex, a fictitious auto parts supplier. It contains three files: *Customer, Order,* and *Part.* The *Customer* file contains the names and addresses of Omnifex's customers. The *Order* file contains orders from these customers. The *Part* file contains information about the parts sold by Omnifex. The Sample Database is also used in examples throughout the *Beginner's Guide,* which provides a tutorial introduction to Cornerstone.

In this *Owner's Handbook,* we have highlighted key concepts with a ☞ pointing hand ☜ in the margin. ☜

☒ Paragraphs printed like this one describe minor details or other points that you may want to skip.

An index is included at the end of each volume in the *Owner's Handbook.*

Although you can read the *Owner's Handbook* from cover to cover (we hope you will), the chapters are more or less self-contained, so feel free to read whatever sections interest you, in whatever order you like. Once you become familiar with Cornerstone, this *Owner's Handbook* and the Quick Reference Card will serve as your primary reference tools.

AN OVERVIEW OF CORNERSTONE

1
CHAPTER

CHAPTER 1

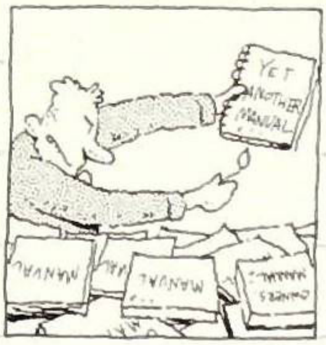
CONTENTS

An Overview of Cornerstone

Welcome to Cornerstone

You think we don't know how much you already have to read? You think we expect you to leap into this manual with a smile on your face? We know better.

This manual was written by people who hate reading manuals too. We've tried our best to make this one readable by keeping it simple. Important concepts are illustrated with clear examples. Less important details are separated out. If you spend some time with this *Owner's Handbook* you will be rewarded with an understanding of the powers of Cornerstone, and the knowledge to use it effectively in your business.



Support for You

Your Cornerstone box comes with two support systems. One consists of all the printed materials: the *Owner's Handbook,* the *Beginner's Guide,* the Quick Reference Card, and the keyboard template. The other system is a built-in manual, our on-line help system. Whenever you're using Cornerstone, you are only one keystroke away from help.

The *Owner's Handbook* describes all of Cornerstone's features. It's more than just an explanation of commands: it's a repository of accumulated wisdom from the developers and early users of the system. The *Beginner's Guide* introduces you to database concepts and teaches you the major features of Cornerstone. While especially useful if your familiarity with computers and databases is limited, it's also a quick way for any user, novice or experienced, to get an overview of Cornerstone.

The built-in support system is accessed via two keys: ⟨HELP⟩ and ⟨OPTIONS⟩. Pressing the ⟨HELP⟩ key explains what you are doing and what you can do next. It's like always having a reference manual open to the right page. ⟨OPTIONS⟩ tells you *exactly* what choices are available from your current position. When you're entering data, ⟨OPTIONS⟩ will tell you about any restrictions that exist and will show you all of the allowed values, even when those values come from other files. The more you work with Cornerstone, the more you'll appreciate this feature.

Getting Started

Everyone has a different style of learning and a different level of experience in using computers, so there's more than one way to use these support materials. However, we do have a few recommendations:

- □ Take a look at everything in the package.
- □ Read the "Read This First" brochure to find out about the installation procedures.
- □ Read the *Beginner's Guide* and work through the tutorial lessons. The few hours this takes will be well worth your time.
- □ If you're too impatient to do that, or if you already have a fair amount of background with databases, you can skip to the "Introduction for the Impatient User," which explains all essential Cornerstone terms, keyboard usage, and what you see on the screen.
- □ Play with the Sample Database. All of the examples in the *Beginner's Guide* and the *Owner's Handbook* refer to this database, which details sales of a fictitious auto parts company, Omnifex.
- □ Explore the Client Tracking system, which has been designed as both a demonstration system and a usable application. It will maintain a Rolodex Desk Top File™ type of directory for individuals and companies, and keep a history of all client contacts. In addition, you can tailor it to meet your specific needs.
- □ Finally, even if you think you know everything about databases, please read the chapter "Planning Your Database." This chapter explains how to go about designing your Cornerstone database.

What Cornerstone Is

Once you've gone through the *Beginner's Guide* or the "Introduction for the Impatient User," you will probably have a good idea of how Cornerstone operates, but it will take some time and practice to understand everything you can do with it. The following application examples should help, but remember, we can't tell you everything Cornerstone will do for you—you will find that out for yourself over time.

At first the only people at Infocom using Cornerstone were those responsible for developing and marketing it. One of the principal developers, who moonlights as our softball coach, built an application that computes (and, unfortunately for some of us, prominently displays) the individual and team batting averages after each game. Another of the main developers built a sophisticated time-billing system for a consulting business.

Then one of the marketing people (whom we can never get off the phone) built a Rolodex system to keep track of the companies and individuals he contacts in the course of his work. Many other people in marketing copied or adapted the same system (which is now part of the Client Tracking system).

Over time, other people within the company came by to look at Cornerstone, and they thought up new uses. Our president started using Cornerstone to keep track of his large science fiction library. Then, after he became more familiar with Cornerstone, he built something a bit more serious: an application for analyzing sales of Infocom's products by type of computer, by product, and by customer. (He gets the raw data from our minicomputer using Cornerstone's Convert utility.)

Our product support representative realized she could use Cornerstone to keep a history of all written customer contacts and to monitor any problems encountered by our interactive fiction customers. Previously this work was done by hand.

Our most sophisticated in-house application was developed for use by our personnel department, replacing a minicomputer time-sharing service with a personal computer and hard disk. Virtually every feature of our old personnel system was replicated, and new features were added as well. Our personnel manager, who has little experience with computers, recently realized the ease of developing new reports and is expanding the functionality of the system. One of her latest reports prints detailed descriptions of all the currently unfilled positions within the company.

The list keeps growing. Our purchasing manager (who had never used a computer before in her life) is starting to use Cornerstone to track purchase orders, keep an inventory of equipment, and monitor equipment maintenance schedules. Our public relations manager keeps a detailed history of her contacts with media organizations. People who travel often on business are using Cornerstone to keep track of their billable expenses. Cornerstone is even used to keep a history of its own testing, with detailed notes recording feedback from testers.

Applications developed outside Infocom have also varied widely. A loan officer for a large New England bank built a "Loan Officer Support System" that monitors all of the paperwork associated with different types of loans. This system is now in use throughout the bank. A wholesale manufacturer of fine chocolates is using Cornerstone to keep track of inventory and monitor machine maintenance. A physician at a major hospital is using Cornerstone to analyze survey results, while his colleague uses it to catalog journal references. The controller of a small metal products manufacturing company uses Cornerstone to do job costing: calculating materials, labor needs, and costs for various production runs.

You get the idea—a wide range of information needs can be addressed by Cornerstone. But don't feel you have to think of everything overnight. Cornerstone has been designed to grow with you. You can add new files, new types of information, and new reports as you go along.

What Cornerstone Is Not

First, Cornerstone is not a spreadsheet. If you use a spreadsheet for doing "what if" and "sensitivity" analyses, you'll probably want to keep it. On the other hand, if you've been using your spreadsheet for storing quantities of data and preparing reports, you now have a more appropriate tool with Cornerstone. And remember, the Convert utility lets you transfer data to or from most popular spreadsheets.

Cornerstone is also not a word processor. While Cornerstone's text-handling capabilities exceed those of other databases, they don't rival most word-processing programs. You could write block style letters with Cornerstone—you could even send form letters that substitute different people's names and addresses. But for most writing you'll want a word processor. One thing to keep in mind: it's easy to include a Cornerstone report in a larger word-processing document.

Where To Go From Here

The *Owner's Handbook* describes every aspect of Cornerstone, but that doesn't mean you need to read each volume from cover to cover. Volume I covers all of the major functions, while Volume II contains more advanced concepts that are less central to everyday use.

The next chapter summarizes the essentials for using Cornerstone. So if you're one of those people who started here without looking at the *Beginner's Guide,* the "Introduction for the Impatient User" is for you. Even if you went through the tutorial lessons, the next chapter is good for refreshing your memory. After that, take the approach that feels right for you. You may want to start using Cornerstone right away, and use the *Owner's Handbook* as a reference manual when necessary, or you may want to read through some or all of the *Owner's Handbook* first. In either case, look at the chapter "Planning Your Database" before building your own database.

We hope you enjoy working with Cornerstone.

INTRODUCTION FOR THE IMPATIENT USER

2 CHAPTER

CHAPTER 2

CONTENTS

Introduction for the Impatient User

Introduction

This chapter covers the minimum you need to know to use the *Owner's Handbook,* and is intended to be a quick alternative to the *Beginner's Guide.* You might also want to use this chapter as a refresher. If this chapter does not meet your needs, use the *Beginner's Guide* instead. It covers the same topics in more depth and introduces additional aspects of Cornerstone.

How to Run Cornerstone

Two-diskette System

To start Cornerstone on a two-diskette system (assuming you've installed Cornerstone; if not, read "Read This First"), boot the system and get to the DOS prompt (A)). If you want to try out the examples in *Owner's Handbook I,* insert the Sample Database Copy diskette into drive A. Otherwise, insert some other Cornerstone data diskette in drive A. Next, insert the Cornerstone system diskette into drive B and enter the DOS command:

```
A)CORNER
```

Hard-disk System

To start Cornerstone on a hard-disk system, boot the system and get to the DOS prompt (C)). (If you don't see the C), type the DOS command: C: and then press ⟨ENTER⟩.)

Change the current directory to the Cornerstone system directory with the DOS command:

```
C)CD \CORNER
```

To run the system with the Sample Database (from which the examples in *Owner's Handbook I* are taken), enter the DOS command:

```
C)CORNER  SAMPLE
```

To run the system with one of the other databases supplied with Cornerstone, or to run it with one of your own databases, follow the CORNER command with the name of the directory that contains the data files you want to use. For example, if you have database files in a subdirectory named ACCOUNTS, enter

```
    C)CORNER ACCOUNTS
```

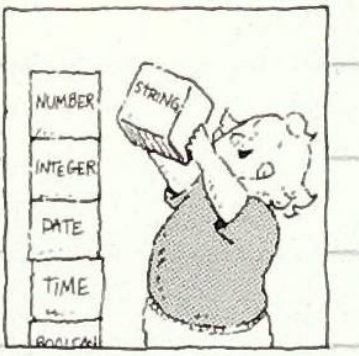
to run Cornerstone with the data in that subdirectory.

Structure of a Cornerstone Database

A *database* is an organized collection of information. In Cornerstone, you organize your database into *files*. Each file contains *records*. A record describes a single person, place, or thing. The Sample Database, for example, is organized into *Customer, Order,* and *Part* files.

The structure of records in a file is determined by the *attributes* you tell Cornerstone the records will have. An attribute is a quality or property of people or things. For example, the records in a file containing information on a fleet of cars might have as attributes the year, make, model, vehicle identification number, and mileage for each car. In the Sample Database, records in the *Customer* file have as attributes the name, address, phone number, and total purchases for each customer.

Because all records in a file must have the same structure, you only define that structure once, when you define the file. What makes one record different from the next are the *values* you enter for attributes. For example, all records in the Sample Database's *Customer* file have the *Name* attribute, but the value for that attribute is different in each record.



In Cornerstone, an attribute can have one of seven *data types:* String, Number, Integer, Date, Time, Boolean, and Enumerated. (For example, an attribute containing alphabetic characters, such as a person's name, would be a String; an attribute referring to an amount of money would be a Number.)

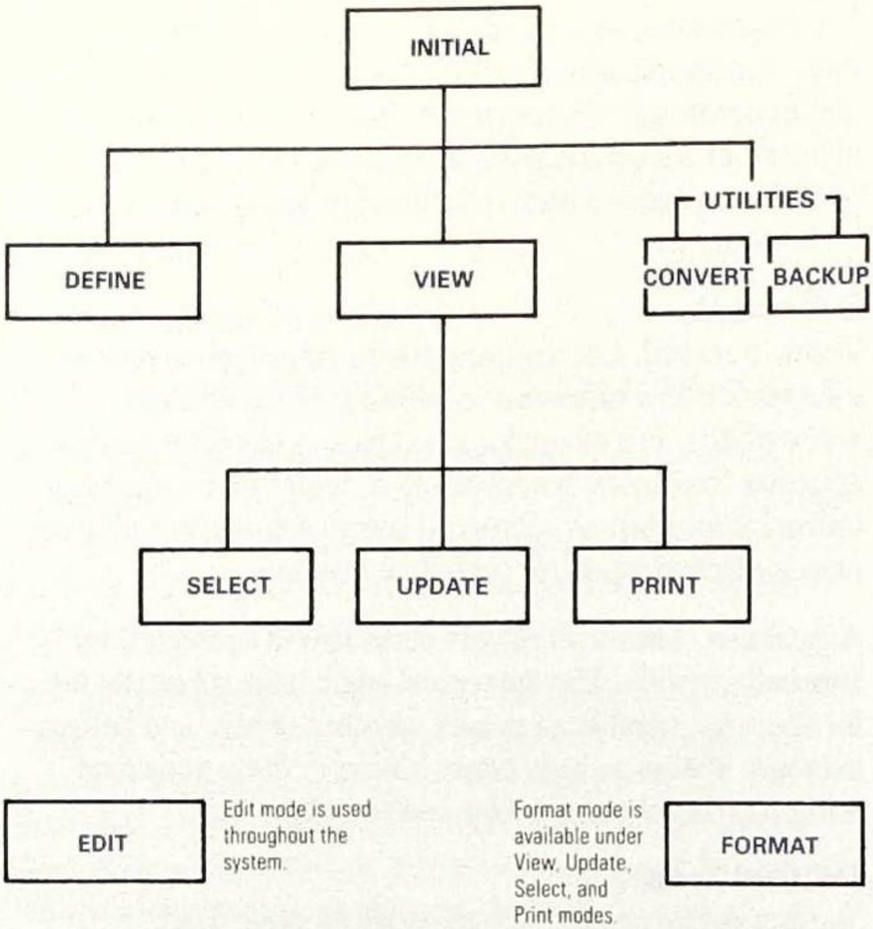
Subrecords
Within a record, Cornerstone allows you to group related information as a *subrecord,* consisting of one or more *subattributes.* For example, a purchase order often has a separate line for each item ordered, with a name, quantity, and price for each. A subrecord allows you to enter all three pieces of information for each item ordered.

A database of baseball players could have a subrecord for baseball seasons. The subrecord might have subattributes for the year, number of at-bats, number of hits, and batting average. Within a single player's record, then, you could enter a group of statistics for each season.

Derived Values
You can define an attribute to *derive* its value from information elsewhere in the file, or even from another file. Derived values are automatically filled in and kept up-to-date by Cornerstone. When you define an attribute to be derived, you supply a *derivation expression,* which tells Cornerstone how to derive the value.

To derive information from one file into another, you must define a *relationship* between the two files. The relationship acts as the path on which the information is sent.

Cornerstone Modes



Edit mode is used throughout the system.

Format mode is available under View, Update, Select, and Print modes.

The use of Cornerstone entails just four basic tasks: entering (and updating) records in your database, selecting the records you want to see, looking at the records you have selected, and printing reports. You will perform several other tasks less frequently, such as designing formats and backing up your database. Cornerstone has a *mode* for each task. Generally, you enter a mode by issuing the command of the same name. (To enter Select mode from View mode, for example, you issue the SELECT command.)

Initial mode is used to start and finish each session with Cornerstone, use the utilities (Convert, Backup, and Recompute), delete formats and selection criteria, issue commands to make new records, and reach Define and View modes.

View mode is used to look at records, organize the display into columns or rows, create reports, sort records, and issue commands to reach Update, Select, Format, and Print modes.

Select mode is used to specify criteria to select records.

Update mode is used to create, change, and delete records.

Edit mode is used to enter or change values in forms.

Print mode is used to set up specifications to print the information displayed in View mode.

Format mode is used to design forms with which to display data.

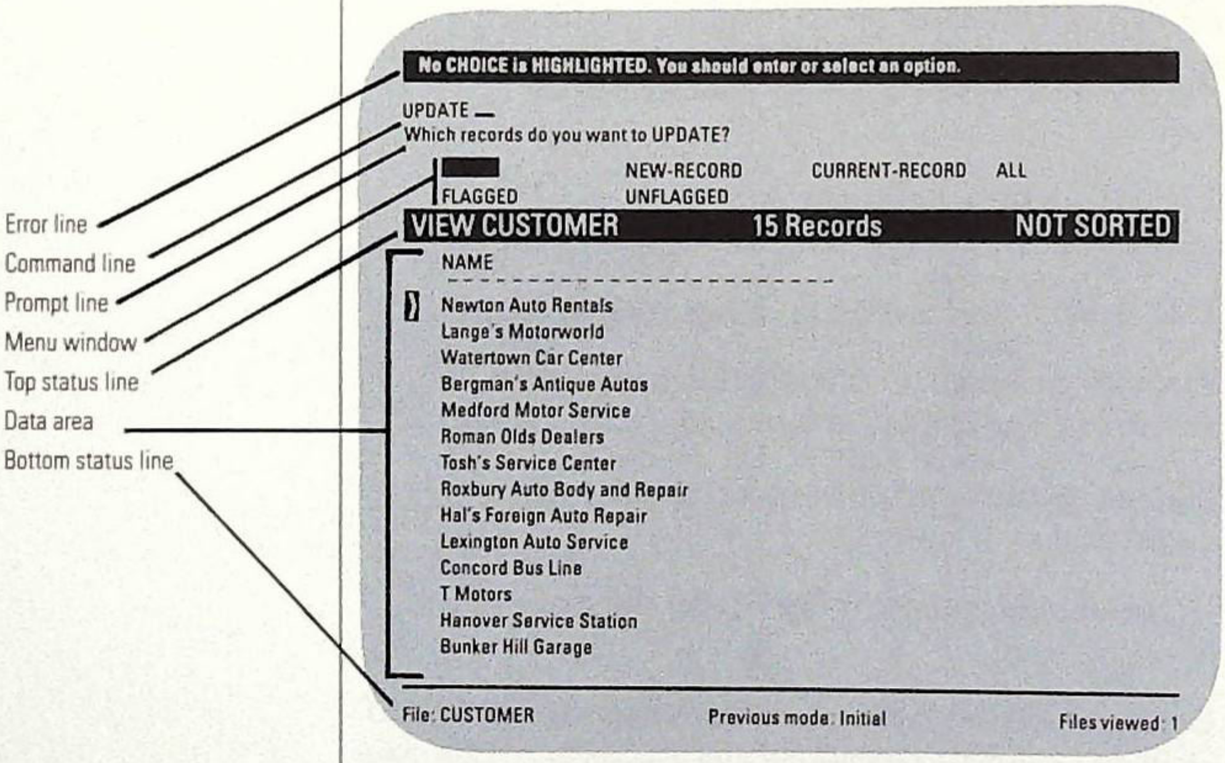
Define mode is used to define or alter the structure of your database.

Convert mode is used to convert data from other products for use in Cornerstone, and vice versa.

Backup mode is used to make backups of your database.

Cornerstone Keys and Screen

Throughout the *Owner's Handbook* and in system messages we refer to keys by their function in Cornerstone, not by the characters or names on the keys themselves. The keyboard template tells you the function of keys F1 through F10, and the Quick Reference Card shows the location of all keys that have a special function in Cornerstone.



```
No CHOICE is HIGHLIGHTED. You should enter or select an option.
UPDATE _
Which records do you want to UPDATE?
                     NEW-RECORD          CURRENT-RECORD     ALL
 FLAGGED             UNFLAGGED
VIEW CUSTOMER                   15 Records                NOT SORTED
 NAME
 -------------------------
 Newton Auto Rentals
 Lange's Motorworld
 Watertown Car Center
 Bergman's Antique Autos
 Medford Motor Service
 Roman Olds Dealers
 Tosh's Service Center
 Roxbury Auto Body and Repair
 Hal's Foreign Auto Repair
 Lexington Auto Service
 Concord Bus Line
 T Motors
 Hanover Service Station
 Bunker Hill Garage

File: CUSTOMER               Previous mode: Initial           Files viewed: 1
```

From top to bottom, the Cornerstone screen consists of:

Error line. If you enter an invalid command or try to enter something into a form that doesn't make sense, Cornerstone will display an error message indicating the problem.

Command line. This line displays the command as you enter it.

Prompt line. The prompt tells you what to do and may briefly explain the options in the menu.

Menu window. This window displays the options to select from.

Top status line. This line tells which mode you're in and gives other information depending on the mode.

Data area. This area, which takes up most of the screen, displays your data. It also displays help and options messages when you press ⟨HELP⟩ or ⟨OPTIONS⟩.

Bottom status line. This line tells you what file you are currently working with, the mode you will return to when you press ⟨DONE⟩, and the number of files you are viewing.

Issuing a Command

A Cornerstone command consists of a command word followed by options. Both command words and options are chosen from the *menu.* You build a command by highlighting a choice from the menu and moving it to the command line. There are two ways to do this:

- □ Use the arrow keys to highlight the option you want, then press ⟨SPACE⟩ to move it to the command line.
- □ Type the option—it will be highlighted as soon as you have typed enough characters to differentiate it from the other options (one or two characters is usually enough). Press ⟨SPACE⟩ to move it to the command line.

When you add an option to the command line, Cornerstone may display a new set of options to pick from. In most cases the *prompt line* tells you to highlight an option in the new menu and add that option to the command line. If you press ⟨ENTER⟩ before the command is complete, a message in the *error line* will tell you that the command can't be executed.

Until you press ⟨ENTER⟩ to execute the command, you can edit the command line by pressing ⟨RUB-WORD⟩, which deletes the last option from the command line; ⟨BACKSPACE⟩, which deletes the last character from the command line; or ⟨CANCEL⟩, which erases the entire command line.

When your command line is complete, a message in the prompt line will tell you to press ⟨ENTER⟩ to execute the command.

Entering Values in a Form

Whenever you have to fill out a form, either Cornerstone will put you in Edit mode automatically or the EDIT command will be available in the menu.

Once you are in Edit mode, the first field in the form is highlighted. Anything you type is entered as a value in the highlighted field. Simply type the value, then press ⟨ENTER⟩ or ⟨NEXT⟩ to store the value and move the cursor to the next field. When you have entered values in as many fields as you want, press ⟨DONE⟩.

Here are some of the keys and control-key combinations you can use to edit a form:

⟨NEXT⟩	Move to the next field
⟨PREVIOUS⟩	Move to the previous field
⟨ARROW-RIGHT⟩	Move to the next character
⟨ARROW-LEFT⟩	Move to the previous character
⟨ARROW-UP⟩	Move to the previous line
⟨ARROW-DOWN⟩	Move to the next line
⟨DONE⟩	Leave Edit mode and return to the previous mode
⟨ENTER⟩	Move to the next field; leave Edit mode (same as ⟨DONE⟩ if you are on last field in form)
⟨CANCEL⟩	Recover the previous value in a field, or clear all values from the form
⟨BACKSPACE⟩	Delete the character to left of cursor
⟨RUB-WORD⟩	Delete the previous word

⟨DELETE-CHAR⟩	Delete the character at cursor
⟨INS/OVS⟩	Change from inserting characters to overstriking, and vice versa
⟨FIRST⟩	Move to the first field in the form
⟨LAST⟩	Move to the last field in the form
⟨OPTIONS⟩	Display a message showing what you can enter in the field
⟨HELP⟩	Display a message describing the current field
Ctrl-A	Move to the beginning of the line
Ctrl-E	Move to the end of the line
Ctrl-K	Delete from the cursor position to the end of the value

If the value you enter is longer than the width of the field, the value will continue on a new line automatically (as in a word processor).

Fields that are defined to take multiple values will expand when you press ⟨ENTER⟩, allowing room for more values. Similarly, when you enter the last value in a subrecord, as soon as you press ⟨ENTER⟩ Cornerstone automatically expands the form to accept another subrecord. If you then continue to press ⟨ENTER⟩ without typing values, Cornerstone erases the empty value or subrecord and moves to the next field.

You can't edit a derived attribute, or one that was defined to be non-editable. The field cursor jumps over these fields instead of resting on them.

Whenever you enter a value into a form, Cornerstone tells you if the value is not allowed. This ensures that values you enter are consistent with your database definition. For example, a field may have one or more constraints that prevent you from entering certain values. You can always press ⟨OPTIONS⟩ and ⟨HELP⟩ to see what values are allowed for the current field.

A value is stored as soon as you move the cursor off the field or press ⟨DONE⟩. If you begin editing a value and then change your mind, you can get back the original value if you haven't moved the cursor off the field. Just press ⟨CANCEL⟩ followed by ⟨ENTER⟩, which executes the CANCEL VALUE command.

Exiting Cornerstone

To end a Cornerstone session, issue the EXIT command in Initial mode. You can get to Initial mode by issuing the ALL-DONE command (where available) or by repeatedly pressing ⟨DONE⟩.

☞ The EXIT command is the only safe way to exit Cornerstone. ☜ Any other way may ruin your data.

The ⟨HELP⟩ Key

You can press the ⟨HELP⟩ key any time. When you are entering a command, pressing ⟨HELP⟩ explains each menu option. When you enter values in a form, pressing ⟨HELP⟩ tells you about the field you're on, and if you press ⟨HELP⟩ a second time, you get tips on editing. If you get an error message while using Cornerstone, press ⟨HELP⟩ to learn what to do. And if you misplace your keyboard template, press ⟨HELP⟩ two or three times to learn the location of Cornerstone command keys and control characters. To clear the Help message from the display, press ⟨CANCEL⟩, ⟨DONE⟩, or ⟨ENTER⟩. Anything else you do will both clear the Help message from the screen *and* be taken as input.

The ⟨OPTIONS⟩ Key

The ⟨OPTIONS⟩ key serves two purposes. When there are more options in the command menu than fit in the menu window, you can press ⟨OPTIONS⟩ to expand the window and see all the options. When you are entering values in a form in Edit mode, pressing ⟨OPTIONS⟩ displays a message telling you what kind of entry you can make in that field.

The Sample Database

Before you jump right in and define the structure of your own database, see what a typical application looks like by experimenting with the Sample Database. The database is for a fictitious car parts company called Omnifex. It has three files: a *Customer* file, which contains information about each customer; an *Order* file, which contains the orders placed by Omnifex's customers; and a *Part* file, which contains records for each part and tool in the Omnifex catalog.

When you have become familiar with the Sample Database, read chapter 9, "Planning Your Database," before you go on to define your own database or convert data to a Cornerstone database.

Don't Pull the Plug ☞

☞ Leave the system only with the EXIT command. ☜ If you mistakenly begin an operation that takes a long time to complete, you can press Ctrl-Break (hold down Ctrl and press the Break key) to safely stop the operation and exit Cornerstone. Any other means of ending a session will ruin your data.

☞ Make back-up copies of your files regularly with the Backup utility. ☜ If your diskettes or hard disk are physically damaged, if you experience a power failure while using Cornerstone, or if you exit Cornerstone improperly, you may ruin your database. Be sure you always have a back-up copy.

Don't attempt to manipulate your Cornerstone files with DOS commands: you will ruin your data.

☞ Never remove a diskette from a drive at any time during your Cornerstone session, even if the drive light is off and the diskette is not spinning. ☜ Remove a diskette only after you have left Cornerstone and the DOS command prompt has re-appeared. (Exception: you may remove diskettes during the Backup process when told to do so.)

VIEWING YOUR DATA

3 CHAPTER

CHAPTER 3

CONTENTS

Viewing Your Data

Introduction

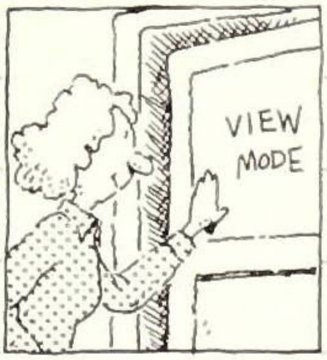
In this chapter you'll learn how to look at the information in your files. With Cornerstone, you can design an unlimited variety of reports for the screen or printer, displaying exactly what you want to see, where you want to see it.

View Mode

In View mode you create new reports in a matter of seconds, displaying some or all of your data. You can then save these reports for repeated use.

In View mode you can also jump from one file to another. While you're busy looking at the orders of your best customers, you might get an urgent call: someone needs to check the price of a new part. Without losing your place in the *Order* file, you can view the *Part* file, answer the question, and return to your work in the *Order* file.

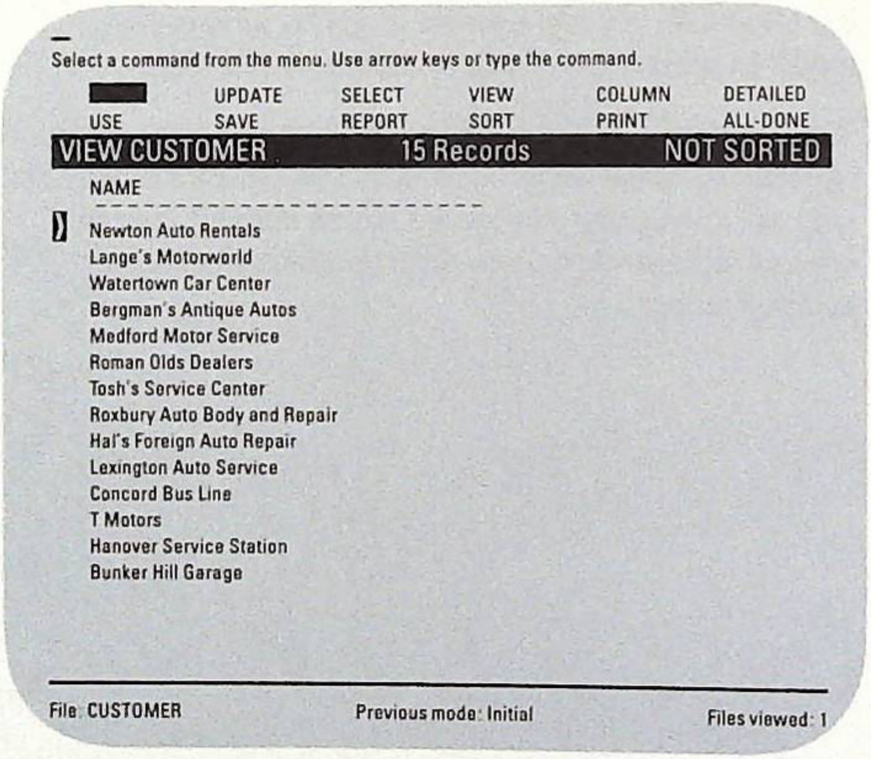
View mode is also the starting point for many other database operations. These include adding or changing information in your files, selecting only those records meeting specified criteria, developing custom-tailored report formats, and printing reports.



Entering View Mode

To enter View mode, select the VIEW command from the Initial menu. Cornerstone then displays a menu showing the names of your files and subfiles. You must select one to complete the command.

The figure below shows the screen display after entering View mode with the *Customer* file from the Sample Database. The menu area displays the available commands from View mode, the status lines tell you about your current position, and the data area shows some of the information contained in the *Customer* file. To the left of the first customer name is the *record cursor* indicating the *current record*, the record the system is ready to work on.



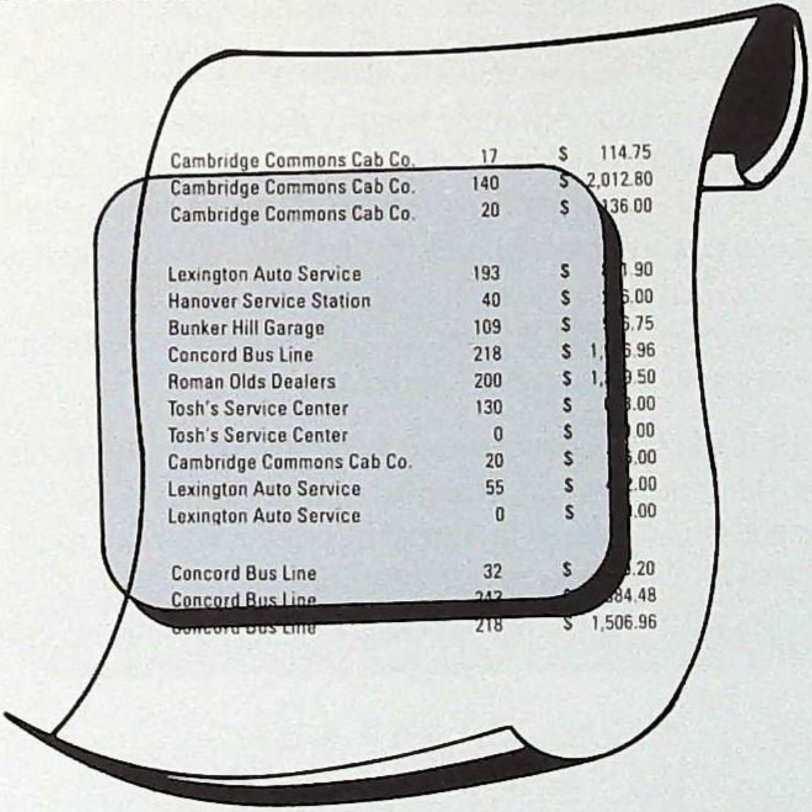
```
_
Select a command from the menu. Use arrow keys or type the command.
              UPDATE      SELECT      VIEW       COLUMN      DETAILED
    USE       SAVE        REPORT      SORT       PRINT       ALL-DONE
VIEW CUSTOMER             15 Records                       NOT SORTED
    NAME
    -------------------------
)   Newton Auto Rentals
    Lange's Motorworld
    Watertown Car Center
    Bergman's Antique Autos
    Medford Motor Service
    Roman Olds Dealers
    Tosh's Service Center
    Roxbury Auto Body and Repair
    Hal's Foreign Auto Repair
    Lexington Auto Service
    Concord Bus Line
    T Motors
    Hanover Service Station
    Bunker Hill Garage

File: CUSTOMER              Previous mode: Initial           Files viewed: 1
```

The way your information is displayed on the screen is the current *report format.* In this case, the report format consists of a column of customer names with a header, *NAME*, describing this information.

To leave View mode, press the ⟨DONE⟩ key or select ALL-DONE from the menu.

Scanning Your Data

You enter View mode looking at all the records from the file you've specified. However, the records may not all fit on the screen at once. You can think of the display as a scroll with the records printed one after another, and the screen as your window on the scroll. The entire scroll is as long as all the records, and can be up to 255 characters wide. The screen is of course much smaller, so it generally shows only part of the scroll.



Cambridge Commons Cab Co.	17	$	114.75
Cambridge Commons Cab Co.	140	$	2,012.80
Cambridge Commons Cab Co.	20	$	136.00
Lexington Auto Service	193	$	.90
Hanover Service Station	40	$	6.00
Bunker Hill Garage	109	$	6.75
Concord Bus Line	218	$	1, 6.96
Roman Olds Dealers	200	$	1, .50
Tosh's Service Center	130	$	8.00
Tosh's Service Center	0	$	.00
Cambridge Commons Cab Co.	20	$	.00
Lexington Auto Service	55	$	.00
Lexington Auto Service	0	$	.00
Concord Bus Line	32	$	.20
Concord Bus Line	242		84.48
Concord Bus Line	218	$	1,506.96

The group of records that you're working with is called the *working set.* The working set initially consists of all the records in the file, but you can restrict the working set to just records you're interested in. (See chapter 5, "Selecting Records.")

There are two basic ways to move through your report:

□ You can move the *record cursor* with any of the motion keys: ⟨NEXT⟩, ⟨PREVIOUS⟩, ⟨FIRST⟩, or ⟨LAST⟩.

□ You can move the *scroll* with any of the scrolling keys: ⟨PAGE-FORWARD⟩, ⟨PAGE-BACKWARD⟩, ⟨SCROLL-FORWARD⟩, or ⟨SCROLL-BACKWARD⟩.

Changing the Appearance of Your Data ☞

Cornerstone lets you view the information from your files in an unlimited number of display formats. You should feel free to change and to experiment with formats. ☞ Changing the format has no effect on the data in your files. ☜

Cornerstone provides three ways to display data: columnar formats created with the COLUMN command, row-oriented formats created with the DETAILED command, and custom-tailored formats created with the REPORT FORMAT command. The sections below describe the COLUMN and DETAILED commands. Chapter 8, "Designing Forms," describes the use of the REPORT FORMAT command.

Producing Columnar Reports

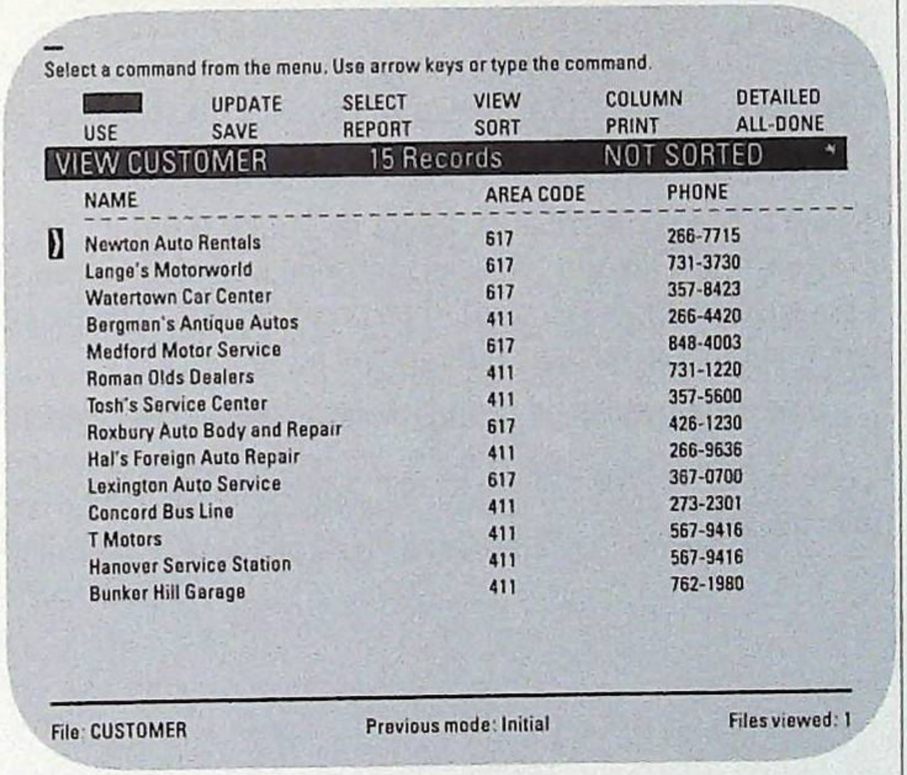
The COLUMN command is used to create columnar reports that show one or more attributes from a file. Each attribute you select is displayed in a column, and the attribute name appears in the header above the column.

For example, suppose sales have been falling off lately and you've decided to call your customers to remind them that you're still in business. Using the *Customer* file, issue the command

COLUMN NAME AREA_CODE PHONE

(If there are more attributes in the file than can be displayed in the menu, just press the ⟨OPTIONS⟩ key to see all the choices.) Once you've selected all the attributes you're interested in, press ⟨ENTER⟩.

That's it—the report you wanted appears on the screen, as shown below.



```
Select a command from the menu. Use arrow keys or type the command.
             UPDATE     SELECT     VIEW       COLUMN     DETAILED
     USE     SAVE       REPORT     SORT       PRINT      ALL-DONE
VIEW CUSTOMER          15 Records            NOT SORTED
     NAME                              AREA CODE      PHONE
     -------------------------------------------------------------
}    Newton Auto Rentals               617            266-7715
     Lange's Motorworld                617            731-3730
     Watertown Car Center              617            357-8423
     Bergman's Antique Autos           411            266-4420
     Medford Motor Service             617            848-4003
     Roman Olds Dealers                411            731-1220
     Tosh's Service Center             411            357-5600
     Roxbury Auto Body and Repair      617            426-1230
     Hal's Foreign Auto Repair         411            266-9636
     Lexington Auto Service            617            367-0700
     Concord Bus Line                  411            273-2301
     T Motors                          411            567-9416
     Hanover Service Station           411            567-9416
     Bunker Hill Garage                411            762-1980

File: CUSTOMER              Previous mode: Initial          Files viewed: 1
```

The column format is the easiest way to look at just a few attributes from a file. If the attribute values are not extremely long, each record will fit on one or two lines, so several records will fit on the screen at once. You can use the scrolling and motion keys to scan your report beyond the first screenful of information.

☒ You can use Format mode to change the position of the columns or the header (or both), change (or eliminate) the header names, or insert additional columns into an existing report. The width of your columns is determined by the default display widths established when you defined the attributes.

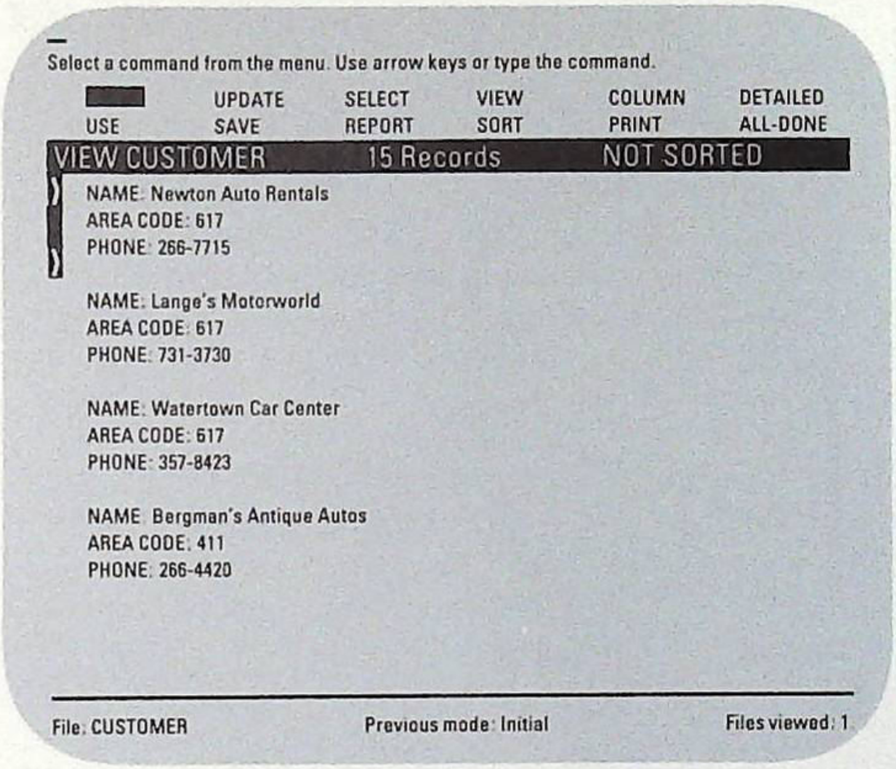
Producing Detailed Reports

Although the columnar format is handy for viewing a few attributes, the screen becomes crowded as more attributes are displayed. When this happens try using the DETAILED command, which displays each attribute on a separate line. DETAILED works much the same as COLUMN, with one minor difference: The DETAILED command gives you a choice of showing SOME-ATTRIBUTES or ALL-ATTRIBUTES from the file.

To see only some attributes, select DETAILED SOME-ATTRIBUTES. This lets you select attributes from the menu. Choose as many as you want, then press ⟨ENTER⟩. The figure below shows the results of the command

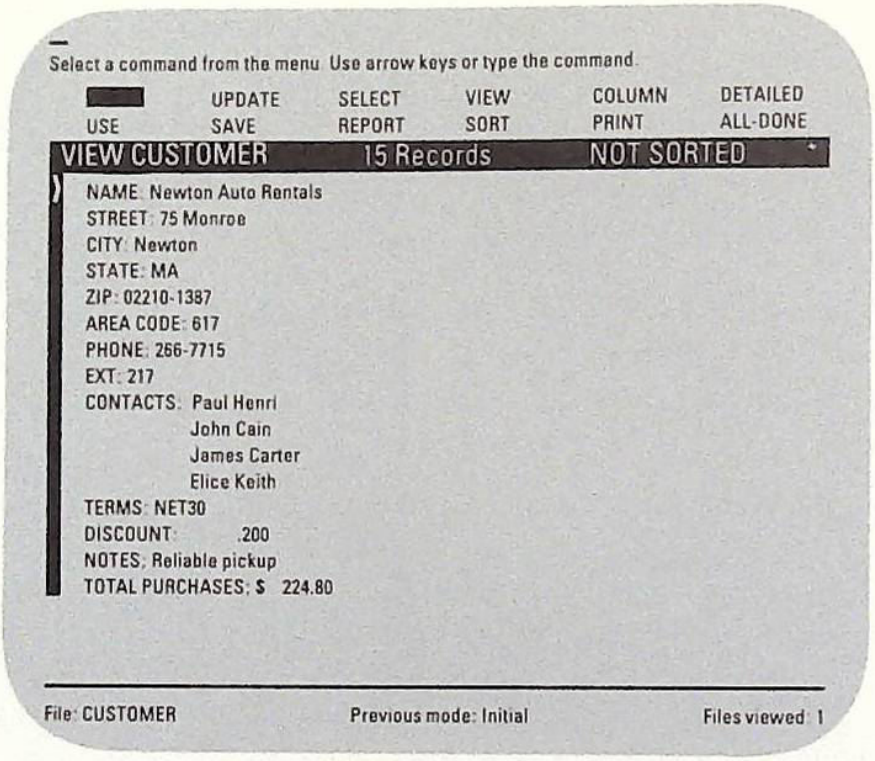
```
DETAILED SOME-ATTRIBUTES NAME AREA_CODE PHONE
```

Compare this to the column format containing the same information.



```
Select a command from the menu. Use arrow keys or type the command.

VIEW CUSTOMER              15 Records              NOT SORTED
   NAME: Newton Auto Rentals
   AREA CODE: 617
   PHONE: 266-7715

   NAME: Lange's Motorworld
   AREA CODE: 617
   PHONE: 731-3730

   NAME: Watertown Car Center
   AREA CODE: 617
   PHONE: 357-8423

   NAME: Bergman's Antique Autos
   AREA CODE: 411
   PHONE: 266-4420

File: CUSTOMER              Previous mode: Initial              Files viewed: 1
```

To see every attribute in the file, select DETAILED ALL-ATTRIBUTES. If you press ⟨ENTER⟩ at this point, all the attributes appear in the order of their definition. To change that order, press ⟨SPACE⟩ before pressing ⟨ENTER⟩ and a menu of attribute names will appear. Select only those attributes you want moved to the top of the report. All remaining attributes will be displayed following the ones you select. A detailed display of all attributes from the *Customer* file is shown below.



```
Select a command from the menu  Use arrow keys or type the command.
              UPDATE      SELECT      VIEW        COLUMN      DETAILED
  USE         SAVE        REPORT      SORT        PRINT       ALL-DONE
 VIEW CUSTOMER          15 Records           NOT SORTED
 ) NAME: Newton Auto Rentals
   STREET: 75 Monroe
   CITY: Newton
   STATE: MA
   ZIP: 02210-1387
   AREA CODE: 617
   PHONE: 266-7715
   EXT: 217
   CONTACTS: Paul Henri
             John Cain
             James Carter
             Elice Keith
   TERMS: NET30
   DISCOUNT:        .200
   NOTES: Reliable pickup
   TOTAL PURCHASES: $  224.80

File: CUSTOMER               Previous mode: Initial               Files viewed: 1
```

You can switch freely between COLUMN and DETAILED formats. Information in a COLUMN format lets you compare records easily. The DETAILED format lets you see more information.

You can store any report format for later use. In working with your database, you will probably design certain commonly-used formats for different needs. For example, you might have one format to display cumulative sales by customer, and another format to print mailing labels for these customers. With Cornerstone you can create a format once, save it, and recall it whenever you need it again.

To save a format, issue the command

```
SAVE FORMAT
```

Cornerstone prompts you to enter a name for the format. You can enter any name you want. Since Cornerstone allows you to use as many as 32 characters in a format name, these names can be quite descriptive. This makes it easier to use your database. The name *Active_Customer_Accounts* tells more about the format than *AcCustAc*. (Use underbars (_) or hyphens (–) instead of spaces in format names.)

Suppose you want to create a format called *Cumulative_Sales* that shows the total number of orders and total value of purchases from each customer. After viewing the *Customer* file, issue the commands

```
COLUMN NAME NUMBER_ORDERS TOTAL_PURCHASE
SAVE FORMAT CUMULATIVE_SALES
```

The first command creates the format. The second command saves it for later use.

Once you save a format, subsequent changes are saved only if you issue the SAVE command again. You can save the changed format under the original name (in which case the original format is overwritten) or under a new name (if you want to keep the original format).

Using Saved Formats

To use a saved format, select USE FORMAT from the menu. Then choose one of the formats you've previously saved from the menu.

☞ A format is tied to a particular file. ☜ You can USE a format for a file only if you saved it in that file.

☒ Whenever you decide to use a new format, Cornerstone checks to see whether the current format has been saved. If it has not, Cornerstone warns you, and gives you an opportunity to save it.

If you name a format *View,* the system will automatically use it whenever you enter View mode. For example, you might want to see the names and phone numbers of your customers every time you enter View mode. To establish this as the default display format you need only issue the following commands

```
COLUMN NAME PHONE
SAVE FORMAT VIEW
```

☒ If you don't have a format named *View,* Cornerstone uses a column format showing the first (non-subrecord) attribute.

If you created a format, intending to keep it for future use, but forgot to issue the SAVE FORMAT command, try issuing the command USE FORMAT PREVIOUS-VIEW. Cornerstone automatically keeps the last unsaved format used in View mode under the name *Previous-View.*

When you save a format in View mode, Cornerstone saves more than just the record display information. The format can also contain

□ sorting instructions

□ headers, subtotals, and totals (see chapter 6, "Advanced Reporting")

□ printing instructions (see chapter 7, "Printing")

You save all this information with the SAVE command and restore it with the USE command.

☒ To delete formats you no longer need, select DELETE FORMAT from Initial mode, and then select the format to delete.

Sorting

Often you will want to arrange the records in your report in a particular order. You might want to alphabetize by customer name, or arrange customers from oldest to newest, from largest to smallest purchasers, or by last purchase date. You might even want to arrange your records on the basis of more than one attribute. (A telephone directory, for example, is ordered by last name and, within the same last name, by first name.)

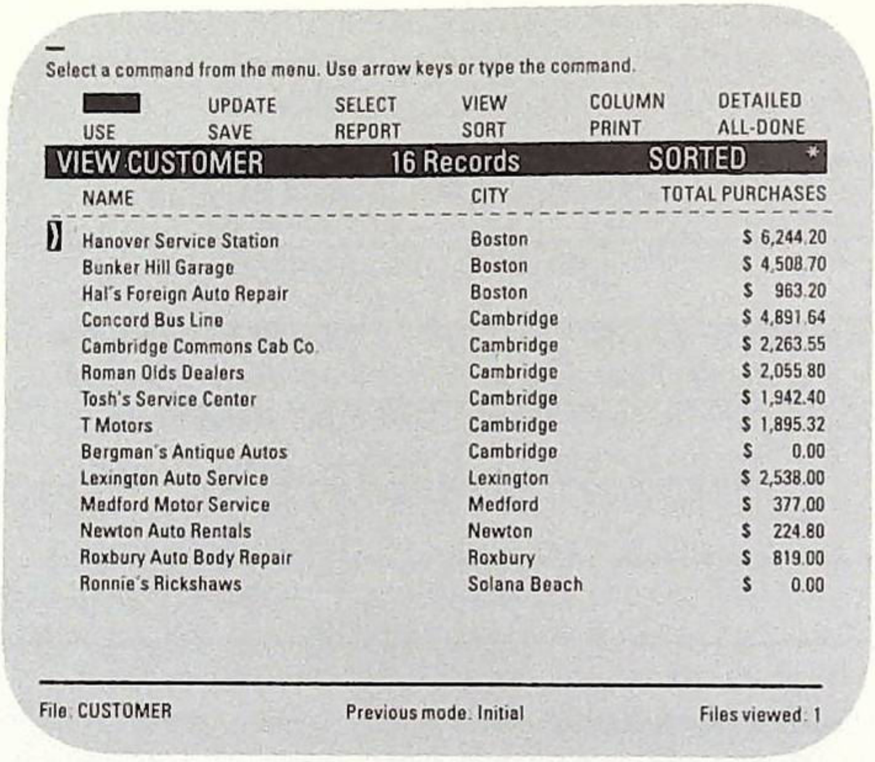
With Cornerstone you can sort your records on the basis of any non-subrecord attribute. Cornerstone imposes no limit on the number of attributes you can sort on.

To sort your records, select the SORT command from the View mode menu. Cornerstone then displays a menu of attributes for you to choose from. After selecting an attribute, you must select a direction: ASCENDING (a to z, lowest to highest number) or DESCENDING (z to a, highest to lowest number). Once you've selected an attribute and a direction, the menu of attributes appears again, allowing you to enter additional attributes and directions. You can select as many attributes as you wish, but one, two, or three suffice for most purposes.

Suppose you plan to take a trip around the country and you want to call on your best customers in each city. You could probably use a report grouping customers by city, with customers from each city ordered from largest to smallest purchasers. To do this, issue the commands

```
COLUMN NAME CITY TOTAL_PURCHASES
SORT CITY ASCENDING TOTAL_PURCHASES DESCENDING
```

The figure below shows the result of issuing these commands. The right corner of the top status line indicates that the current format is sorted.



```
Select a command from the menu. Use arrow keys or type the command.

VIEW CUSTOMER              16 Records              SORTED    *
NAME                                  CITY            TOTAL PURCHASES
----------------------------------------------------------------------
) Hanover Service Station             Boston              $ 6,244.20
  Bunker Hill Garage                  Boston              $ 4,508.70
  Hal's Foreign Auto Repair           Boston              $   963.20
  Concord Bus Line                    Cambridge           $ 4,891.64
  Cambridge Commons Cab Co.           Cambridge           $ 2,263.55
  Roman Olds Dealers                  Cambridge           $ 2,055.80
  Tosh's Service Center               Cambridge           $ 1,942.40
  T Motors                            Cambridge           $ 1,895.32
  Bergman's Antique Autos             Cambridge           $     0.00
  Lexington Auto Service              Lexington           $ 2,538.00
  Medford Motor Service               Medford             $   377.00
  Newton Auto Rentals                 Newton              $   224.80
  Roxbury Auto Body Repair            Roxbury             $   819.00
  Ronnie's Rickshaws                  Solana Beach        $     0.00

File: CUSTOMER          Previous mode: Initial          Files viewed: 1
```

Certain operations, such as adding new records or selecting all the records, can disturb the sorted order. If the sorted order has been disrupted (if the status line says NOT SORTED), you can re-sort by issuing the SORT command without having to respecify attribute names. Cornerstone remembers the last sort command.

☒ After certain commands and database operations, Cornerstone cannot tell if the records are still sorted. To be on the safe side, it assumes that the records are no longer sorted. For example, when you add a new record, it is automatically placed at the end of the report. If the record belonged at the end (Zbigniew Zelany's record, for instance) the order is actually correct. Sort again if you're not sure.

☞ When you save a sorted report format (when the status line says SORTED) you also save the sorting information, both attributes and directions. ☜ When you later use the format, the system automatically sorts the records before displaying them, including any new records that were created after you saved the format.

Even your default View mode format (the one named *View*) can be sorted. Sorting may take some time. How much time depends on the size of your file and your computer hardware. If your file is relatively small, you may prefer a sorted default View mode format; if the file is large you may prefer an unsorted default format. To change a format from sorted to unsorted, issue any command that disrupts the sorted order (such as SELECT ALL) and then save the format under the previous name. If a sort is taking a long time and you want to terminate it before its completion, just press Ctrl-Break (hold down Ctrl and press the Break key) to exit Cornerstone.

☒ If you sort on the basis of a multi-valued attribute Cornerstone uses just the first value. If you want to sort on the basis of a subattribute in a subrecord, you must first use one of Cornerstone's aggregate functions to extract a single value. Consult the chapter on "Entering Information Into Forms" in *Owner's Handbook II* for details.

☒ Any of Cornerstone's basic data types can be sorted. For most, the meaning of ASCENDING or DESCENDING is straightforward. The one data type where the results may be surprising is Enumerated. Cornerstone sorts enumerated attributes in the order in which the values were defined.

Viewing Other Files

Suppose you're looking at the *Customer* file when the phone rings. There's been a mix-up with the orders for Tosh's Service Center, and the folks down in accounting want you to check Tosh's old orders. You could press ⟨DONE⟩, returning you to Initial mode, and then view the *Order* file. However, you're in the middle of creating a new report for the *Customer* file and don't want to lose your position. The better approach is to use the VIEW command from the View mode menu.

From View mode, you can move to any other file with the VIEW command. When you finish with the second file, pressing ⟨DONE⟩ returns you *to your original position* in the previous file. If you like, instead of returning to your original file, you could move to a third file. (Up to eight files can be viewed concurrently in this fashion. The bottom status line displays the number of viewed files.)

To view another file, select VIEW, and then the file you want to view. If the file you choose is related to the current file, you have two options: You can view ALL of the records in the file, or you can view ONLY-RELATED-RECORDS in the file. (If the new file is not related, Cornerstone will display all records in that file.)

If you choose to view only the related records, you will be asked to select one of the following options:

□ **CURRENT-RECORD.** Selects all records related to the current record.

□ **FLAGGED.** Selects all records related to any of the flagged records (flagged with the ⟨FLAG/UNFLAG⟩ key).

□ **UNFLAGGED.** Selects all records related to any of the unflagged records.

□ **ALL.** Selects all records related to any of the records in the current working set.

To look at Tosh's orders, you could enter

```
VIEW ORDER
ALL-RECORDS
```

This tells Cornerstone that you want to view all records in the *Order* file. You would then have to select the records for Tosh's Service Center. If Tosh's record in the *Customer* file was your current record, you could enter

```
VIEW ORDER
ONLY-RELATED-RECORDS CURRENT-RECORD
```

This would place you in View mode with the *Order* file, looking at only Tosh's order records.

☒ In general, all of the View mode commands are available to you whether you have viewed one file or many files, but you cannot update records in a file you are viewing more than once. To update such a file, you must press the ⟨DONE⟩ key until you reach the original occurrence of that file, and then enter Update mode.

Viewing Subfiles

You can directly view a subfile (file of subrecords) as you would any other file. In the Sample Database, for example, you can view the subfile *Line Items*. While viewing a subfile you can change the display format in any way you choose.

☒ You cannot change the information in a subfile directly. You can only update the subrecords in a subfile by updating the records in the parent file. For example, you can only update *Line Items* by updating the line items in the individual *Order* records.

Returning Home

When you have been viewing more than one file, pressing the ⟨DONE⟩ key returns you to the immediately preceding mode. Repeated use of ⟨DONE⟩ will return you to earlier and earlier modes. If you've finished viewing all the files, it's quicker to select ALL-DONE from the menu, which completes all activity and returns you to Initial mode.

The View Mode Menu

You've probably noticed that many choices in the View mode menu have yet to be discussed. Each of the following commands is described in a separate chapter:

UPDATE. Used for actual manipulation of data: entering records, deleting records, and changing records. See chapter 4, "Entering and Updating Records."

SELECT. Used to narrow your working set of records to those meeting specified criteria, such as customers in Cleveland who have never gone to a Cleveland Indians game. See chapter 5, "Selecting Records."

REPORT. Used to prepare more complex reports containing subtotals, totals, and headers, and to design free-form report formats. See chapter 6, "Advanced Reporting."

PRINT. Used to print your report or write it to a disk file. See chapter 7, "Printing."

ENTERING AND UPDATING RECORDS

4

CHAPTER

CHAPTER 4

CONTENTS

Entering and Updating Records

Introduction

Once you have defined your database, you'll want to enter some records into it. And sooner or later you'll want to make changes to records you have entered. Update mode allows you to add new records to your database, change existing records, and delete records. You enter Update mode by issuing the UPDATE command in View mode.

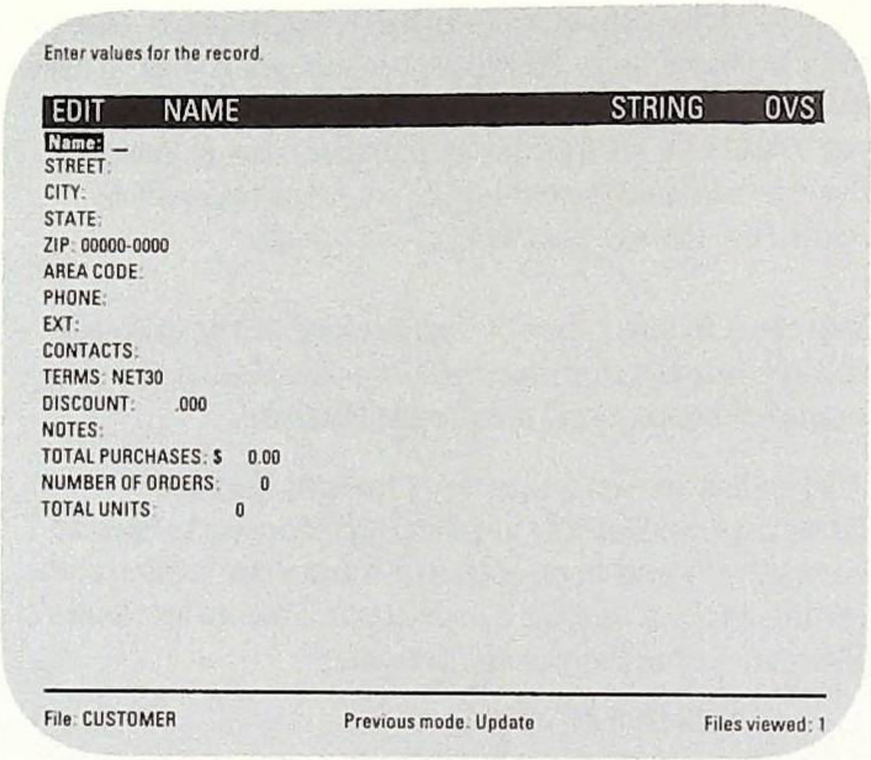
Two Brief Examples

Your most frequent uses of Update mode will be to create new records and to change existing ones. Here is an example of each, using the Sample Database.

A letter has arrived from a new prospect, Ronnie's Rickshaws, announcing the company's decision to open an account with your firm. You are in View mode looking at the records in the *Customer* file. To create a record for Ronnie's Rickshaws from View mode, you enter

```
UPDATE NEW-RECORD
```

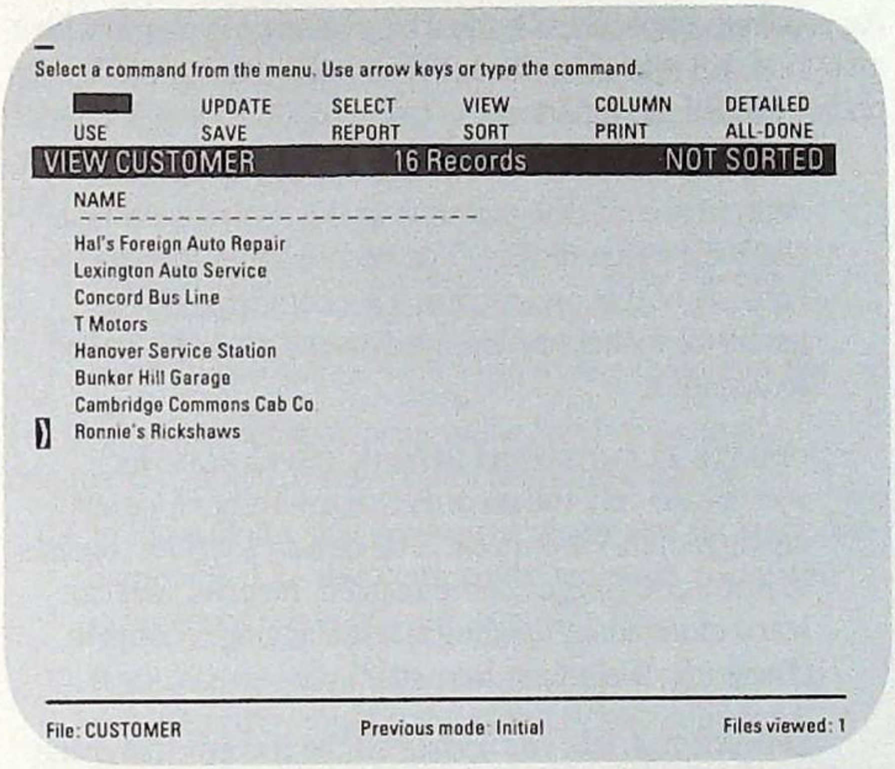
The system displays a new record form, and the first attribute, *Name,* is highlighted. This means Cornerstone is waiting for you to enter a value there. You just type in "Ronnie's Rickshaws," then press ⟨ENTER⟩. This action stores the value and highlights the name of the next attribute, *Street.* Again, you simply type in a value, then press ⟨ENTER⟩.



```
Enter values for the record.

EDIT     NAME                                              STRING     OVS
Name: _
STREET:
CITY:
STATE:
ZIP: 00000-0000
AREA CODE:
PHONE:
EXT:
CONTACTS:
TERMS: NET30
DISCOUNT:       .000
NOTES:
TOTAL PURCHASES: $    0.00
NUMBER OF ORDERS:       0
TOTAL UNITS:            0

File: CUSTOMER                 Previous mode: Update              Files viewed: 1
```

You complete your new record by continuing to type information in each field and pressing ⟨ENTER⟩. When you type a value in the last field and press ⟨ENTER⟩, or when you press ⟨DONE⟩, Cornerstone stores the record automatically. Cornerstone then asks you if you want to create another record. If so, you simply press ⟨ENTER⟩ again. Since you only have one new customer to enter in our example, you answer NO and return to Update mode.

When you press ⟨DONE⟩ again, you return to View mode and see the new record you created for Ronnie's Rickshaws.



```
Select a command from the menu. Use arrow keys or type the command.
              UPDATE       SELECT       VIEW        COLUMN       DETAILED
   USE        SAVE         REPORT       SORT        PRINT        ALL-DONE
VIEW CUSTOMER                16 Records                         NOT SORTED
   NAME
   ------------------------
   Hal's Foreign Auto Repair
   Lexington Auto Service
   Concord Bus Line
   T Motors
   Hanover Service Station
   Bunker Hill Garage
   Cambridge Commons Cab Co
▶  Ronnie's Rickshaws

File: CUSTOMER               Previous mode: Initial            Files viewed: 1
```

When Ronnie's Rickshaws informs you a few months later that they are moving, the record in the *Customer* file can be taken care of just as easily. This time, move the record cursor to Ronnie's Rickshaws, and issue the command

UPDATE CURRENT-RECORD

To change Ronnie's address, use the EDIT command. (You can do this by just pressing ⟨ENTER⟩, since the EDIT command is already highlighted.) Press ⟨ENTER⟩ a couple of times to move the form cursor to the *Street* field, and type in the street address. Pressing ⟨ENTER⟩ again stores the new value and moves the cursor to the next field, '*City,* which also must be changed. Finally, you press ⟨DONE⟩ to leave Edit mode and return to Update mode.

The UPDATE Command

As you have just seen, the UPDATE command is paired with an option that specifies the set of records you want to update, called the *update set.*

> **UPDATE CURRENT-RECORD** is used to update the current record (the one next to the record cursor) in the View mode display. The update set in this case consists of that one record. This command is also useful for examining a record even if you don't intend to update it.
>
> **UPDATE FLAGGED** and **UPDATE UNFLAGGED** let you update only the records that are flagged, or left unflagged, in View mode. The update set then consists of just those flagged, or unflagged, records. You can learn more about flagging and unflagging records in chapter 5, "Selecting Records."
>
> **UPDATE ALL** lets you update all the records that make up your current working set. These records then constitute your update set. (If you have used selection, the working set will be a subset of all the records in the file.)
>
> **UPDATE NEW-RECORD** lets you add one or more new records to your file. This command displays a new form where you can fill in the values for your new record. The update set in this case consists of the new records you create.

Only one record at a time is displayed in Update mode. Whenever you have more than one record in the update set, you can move from record to record by using the ⟨NEXT⟩, ⟨PREVIOUS⟩, ⟨FIRST⟩, and ⟨LAST⟩ keys in Update mode.

You issue the EDIT command in Update mode to change values in an existing record. (You can issue the EDIT command merely by pressing ⟨ENTER⟩.) You then enter or change values one field at a time, moving from one field to the next.

☒ Cornerstone will not let you update records in a file which you are viewing at another level. For example, if you view the *Customer* file, then view another file, then view the *Customer* file again, the system will not let you then enter Update mode. To make changes in the file, press ⟨DONE⟩ until you return to the first level of interaction with that file.

Creating Records

For your convenience, Cornerstone lets you issue a command to create a new record from three places.

- □ In View mode, you can issue the UPDATE NEW-RECORD command. Use this command if you want to create a new record while you are viewing a file.
- □ If you are already in Update mode, you can create a new record by issuing the NEW-RECORD command.
- □ You can also issue the NEW-RECORD command in Initial mode. This command requires you to select the file you want to create records in.

Whichever way you choose to create a new record, the result is the same: Cornerstone puts you in Edit mode with a new record form to fill out.

Entering Values in a New Record

When you create a new record, you enter values into a form. As with any other form in Cornerstone, you need only type a value in the field, then press ⟨ENTER⟩ to store the value. The cursor then moves to the next field, except when

- □ The present field can take multiple values. When you press ⟨ENTER⟩ the field expands to accept another value. If you press ⟨ENTER⟩ again without entering another value, the field contracts and the cursor moves to the next field.

- □ The next field takes a derived value, or was set to be non-editable. A derived value (see Derived and Initial Values, below) is taken from elsewhere in the database—you don't have to enter it. A field is set to be non-editable in Format mode with the SET-CHARAC command (see chapter 8, "Designing Forms"). In both these cases the cursor skips over the field.
- □ You are on the last editable field in the form. In this case pressing ⟨ENTER⟩ is the same as pressing ⟨DONE⟩: you leave Edit mode and return to Update mode.

Before it stores a value, Cornerstone checks to see that the value meets any constraints that may be in effect. If the value doesn't meet a constraint, Cornerstone will prevent you from entering that value. For example, the *Discount* attribute in the *Customer* file of the Sample Database is defined so that you cannot enter a value greater than 1.00. This constraint prevents an entry that would have Omnifex paying its customers to take its goods.

Some of the constraints may have been supplied when you defined the file; others are built into the system. The purpose of constraints is to insure that you enter values consistent with your database. You can always learn exactly which constraints are present for a particular field by pressing the ⟨OPTIONS⟩ and ⟨HELP⟩ keys. Furthermore, you can always change any constraints that were entered when the attribute was defined. Chapter 10, "Defining Your Database," tells you how to create and alter constraints.

Whenever Cornerstone prevents you from making an entry, it tells you why with a specific error message. If you press ⟨HELP⟩ immediately after you get the error message, Cornerstone will explain the problem and suggest how you can correct the situation. You can also press ⟨OPTIONS⟩, and Cornerstone will tell you what is valid in the field. If you must edit the value, you can use the ⟨BACKSPACE⟩, ⟨DELETE⟩, ⟨INS/OVS⟩, and control keys.

There are two built-in constraints that are always present.

- □ Cornerstone won't let you enter a value of the wrong data type. If the field is expecting a date value, like "11/25/84," you can't enter "Fenway Body Shop" or "$19.95."
- □ Cornerstone won't let you enter a value that doesn't make sense. You can't enter "Feb 31, 1984" as a date, for example, or "25:30" as a time.

You may also encounter constraints that were supplied when the file was defined.

- □ Cornerstone will prevent you from entering a value if it falls outside any defined minimum and maximum values. (You can't enter 101 for the attribute *Test Score,* for example, if the attribute is defined to take no value greater than 100.)
- □ When the attribute is defined to take a *unique value,* you won't be able to enter a value that is the same as one already in the file.
- □ You will get an error message if the field is *mandatory* (that is, it must have a value), and you did not type in a value. You won't be able to leave the form until you enter a value or issue the CANCEL FORM command.
- □ The value you are entering may be restricted to be the same as some value in another file. In the Sample Database, for example, a customer name in an *Order* record must equal one of the values for the *Name* attribute in the *Customer* file.

If you are creating many records with the same value for a particular attribute, you can use the COPY command (CTRL-D) to copy the value from the previous record.

☞ Cornerstone automatically deletes empty records. ☜ If you issue a command to create a record but don't enter any values in the form, Cornerstone will delete the record when you leave Update mode. You can delete records that are *not* empty with the DELETE command (see Deleting Records, below).

Derived and Initial Values

Cornerstone may fill in some fields automatically, depending on how you defined your database. The system derives the value from other information in the database, saving you time and eliminating the possibility of error.

Derived Values

In the Sample Database, when you fill in the name of your customer in an *Order* record, Cornerstone checks to see that it matches the name of a customer in the *Customer* file, then takes the address and automatically displays it in the *Order* form. The address in the *Order* file is a set of *derived* fields. A derived field displays a value (or a combination of values) that actually reside elsewhere in the database. You can't edit a derived field—the cursor jumps over it when you are filling in values. If you want to change a derived field, you must change it where it actually resides. In the example above, you would have to change the address fields in the *Customer* file.

Cornerstone is able to display information from the *Customer* file in the *Order* file because a relationship was defined between the two files. The relationship provides the path on which a value in one file is derived from information in another file. You can learn more about relationships and derived attributes in chapter 9, "Planning Your Database."

Initial Values

When you create a new record, some fields may already have values filled in. These are called *initial values,* because Cornerstone automatically fills them in when you create a record, based on an initial value expression you entered when you defined the attribute. An initial value is helpful for the head start it gives you when you create a record. Unlike a derived value (which is merely a "display" of another value or combination of values), an initial value can be edited, because it is actually stored in the record where you see it. The only difference between a value you type in yourself and an initial value is that Cornerstone fills in the value for you when you create the record.

In the *Order* file of the Sample Database, the *Order Number* and *Order Date* fields have initial values that are filled in when the record is created. The *Order Number* field gets its initial value from the UNIQUE_ID function, and the *Order Date* field gets its initial value from the TODAY function.

In the Sample Database, the *Terms* field is *not* filled in when the new record form is first displayed, although it was defined to take its initial value from the related *Customer* record. Terms will differ from customer to customer, so the system doesn't know which terms value to use until you enter the name of the customer. When you identify the customer by entering the customer *Name,* Cornerstone fills in the initial value for the *Terms* field (obtained from the related *Customer* record).

☒ *Terms* is not a *derived* field, because you might want to change a customer's regular terms for a special order. If *Terms* were a derived field, you would only be able to change the *Terms* in the *Order* file by changing the *Terms* in the *Customer* file, and this would change the *Terms* for *all* the orders from that customer.

Updating Initial Values With RECOMPUTE

Normally you write an expression that computes the initial value when you define the attribute in Define mode, but you can change that expression by issuing the INIT-VALUE command in Update mode. For example, the *Order Date* field has an initial value taken from the TODAY function. But suppose you want to enter several orders that were taken yesterday. You can use the INIT-VALUE command to change the initial value expression from TODAY to yesterday's date—say, "1/1/85." Making that one change in the initial value expression saves you from editing the value separately in each record. Now every record you create will automatically have yesterday's date on it. Then, when you have created the records you want with the new initial value, you use the INIT-VALUE command again to change the initial value expression back to TODAY.

☒ Cornerstone fills in an initial value only when the initial value satisfies all the constraints present for the attribute. For example, if an attribute is constrained to be greater than zero, but the initial value expression for a particular record returns a value of less than or equal to zero, Cornerstone will not fill it in.

Cornerstone enters initial values automatically when you create a record. Unless you deliberately edit a value that was filled in by an initial value expression, however, it will not change even if you change the value or expression on which it is based.

To recompute an initial value, use the RECOMPUTE command. Enter Update mode with the record in question, then enter Edit mode. Next, press ⟨ENTER⟩ until the field you want to recompute is highlighted. Return to command level (by pressing the ⟨COMMAND⟩ key), and issue the RECOMPUTE command. This will change the value in that field to one based on current values.

In Edit mode under Update mode the RECOMPUTE command works for one value at a time, one record at a time. To update one or more initial value fields in a group of records, use the UTILITIES RECOMPUTE command in Initial mode. This command lets you specify one or more attributes to recompute, and can recompute an entire file at once.

Creating Subrecords

If your database has subrecords, you create them when you create or update the parent record. As with any other form in Cornerstone, you need only type in a value and press ⟨ENTER⟩ to move to the next field. When you enter the last value in one subrecord, Cornerstone automatically expands the form to accept another subrecord. If you continue to press ⟨ENTER⟩ without typing any values in a subrecord, Cornerstone takes that to mean that you don't want to enter any more subrecords, and stops creating new subrecord forms.

Although you can view records in a subfile directly, you can only *update* a subrecord while you are updating the parent record.

Changing an Existing Record

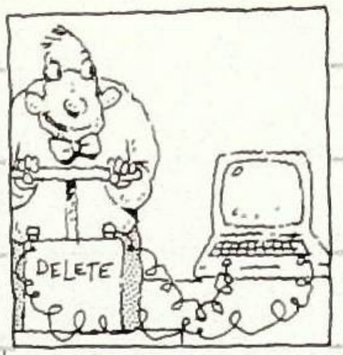
Before you use Update mode to change an existing record, select in View mode the record or records you want to update. In View mode, you can

- □ Move the record cursor to the record you want to update and issue the UPDATE CURRENT-RECORD command.
- □ Flag, or leave unflagged, the records you want to update. You can then issue the UPDATE FLAGGED or UPDATE UNFLAGGED command as appropriate.
- □ Issue the VIEW command again to select the related records of another file, then issue the UPDATE ALL command. ☞ The UPDATE ALL command does not necessarily update all the records in a file. ☜ Rather, it updates all the records in your *working set*. If you used any kind of selection to narrow your working set from the entire file to just a subset of the file, then UPDATE ALL updates just that subset.
- □ Issue the SELECT BY-FORM command and enter some selection criteria, then return to View mode and issue the UPDATE ALL command.

Once you are in Update mode, you can edit the record by pressing ⟨ENTER⟩.

Deleting Records

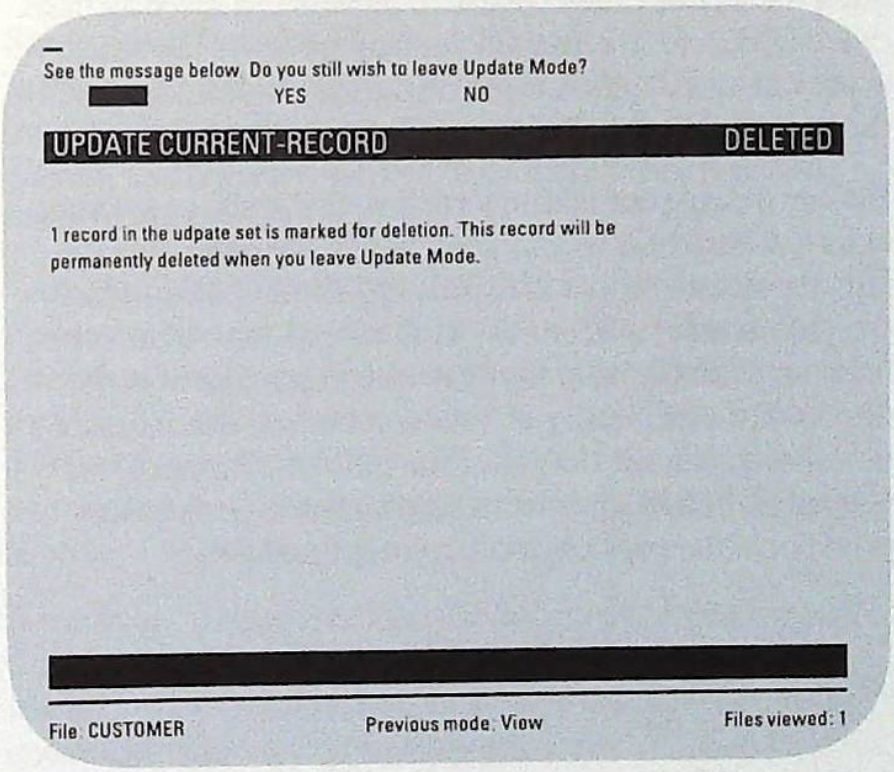
The DELETE command in Update mode lets you delete records. This command takes one of two options. DELETE CURRENT-RECORD deletes the displayed record. DELETE ALL-UPDATE-RECORDS deletes all the records in the update set. When you mark a record for deletion with the DELETE command, the word DELETED appears in the status line for that record. The record is only "marked" for deletion; it is not actually deleted from the database until you leave Update mode.



If you want to delete more than one, but not all, records in your update set, you can delete records individually, moving from one to another with the ⟨NEXT⟩, ⟨PREVIOUS⟩, ⟨FIRST⟩, and ⟨LAST⟩ keys.

If you change your mind and decide you don't want to delete a record, you can issue the UNDELETE command.

If you have marked any records for deletion, Cornerstone won't actually delete them without giving you one last chance to change your mind. Before you leave Update mode, the system reminds you that you have a certain number of records marked for deletion, and asks you to confirm that you want to delete them.



```
See the message below. Do you still wish to leave Update Mode?
                 YES                    NO
UPDATE CURRENT-RECORD                                    DELETED

1 record in the udpate set is marked for deletion. This record will be
permanently deleted when you leave Update Mode.

File: CUSTOMER            Previous mode: View          Files viewed: 1
```

If you answer YES and press ⟨ENTER⟩, the records are permanently deleted from the database: they're gone for good. Answering NO leaves you in Update mode, where you can undelete the records if you want. Simply choosing NO does not undelete the records; you must do this with the UNDELETE command.

☒ Although Cornerstone warns you when it is about to delete a record, you must judge for yourself what effect the deletion may have on your database. Keep in mind that the record you delete may supply values to other records in the database. For example, suppose the Omnifex corporation discontinued part J-1111. If the record for J-1111 were deleted from the *Part* file, the past orders for the part would no longer include a description of it, because the description was stored in the record you deleted.

The Update Format

When you enter Update mode, Cornerstone displays records using the format called *Update*. If there isn't a saved format called *Update* for the file, Cornerstone displays the records in the DETAILED ALL-ATTRIBUTES format, which displays all the attributes in the file, one attribute per line.

You can design your update format so that it shows as many or as few attributes as you want, and in any order. You can limit the update format to include just those fields in which you plan to enter a value. You can't enter a value in a derived field, for example, so you may decide to leave it out of the display. For easier entry of values, you might want to place a field that appears at the end of the DETAILED format at the beginning. You might want to rearrange the fields so they print out in the right places on a pre-printed form.

A quick way to design your own *Update* format is with the DETAILED ALL-ATTRIBUTES command. This command lets you change the order in which attributes are displayed. The DETAILED SOME-ATTRIBUTES command, on the other hand, lets you eliminate attributes from the display. Only the ones you list in the command line will appear in the display. Chapter 3, "Viewing Your Data," tells you more about the DETAILED command.

If you want a more elaborate update form, such as one with more than one attribute on a line, or one with constant text, use the FORMAT command. This command brings you into Format mode, which you can learn more about in chapter 8, "Designing Forms."

Saving and Using an Update Format

You can save any format you design and give it a name with the SAVE command. If you save a format under the name *Update,* Cornerstone will use that format every time you enter Update mode.

You can design other formats in Update mode for special purposes, but you may not want to see them automatically. Save these formats under names other than *Update,* using a name that indicates its purpose, such as *Invoice* or *Label.* To display your records in such formats, issue the USE command with the name you gave to the format when you saved it: USE INVOICE.

If you forget to save a format you designed in Update mode, Cornerstone saves it for you automatically and gives it the name *Previous-Update.* Cornerstone saves only one format under that name at a time, however.

Tips for Designing Update Forms

When you create a format that you plan to call *Update,* make sure it includes all mandatory fields. After all, you can only enter a value in a field if the field is present! You can learn which fields are mandatory by looking at the file definition in Define mode. Also make sure that your *Update* format displays field names, since a new record form without field names is totally blank. The COLUMN command creates a format without field names, so avoid it when designing your *Update* format.

Printing the Update Set

You can get a printout of a single record or all the records in your update set by using the PRINT command. The PRINT CURRENT-RECORD command prints the record currently displayed. The PRINT ALL-UPDATE-RECORDS command prints your entire update set. You might want to use this command if you just entered several new records or edited some old ones and want printouts of them.

SELECTING RECORDS

5

CHAPTER

CHAPTER 5

CONTENTS

Selecting Records

Introduction

When you view a file from Initial mode, your working set consists of all the records in the file. But often you want to work with only some of the records in a file. For example, you might want to write a report on March orders only, or look at your West Coast customers, or pick out the records of people who telephoned you this morning. Cornerstone gives you several ways to narrow your working set to just the records you want.

Two Brief Examples

One of the most powerful ways to select records is to make entries in a selection form. Here are two examples that use the Sample Database.

Suppose you want a list of your customers in the city of Cambridge. Since the View mode display shows only the customers' names, you issue the COLUMN command to show the name and city of each customer:

```
COLUMN NAME CITY
```

Next, you issue the command

```
SELECT BY-FORM
```

to enter Edit mode under Select mode, where you get a selection form to fill out. The values you enter in this form are used to select records for viewing when you return to View mode. In this case you want to select all customers who are in Cambridge, so you move the cursor to the *City* field and type in the name of that city:

```
CITY: Cambridge
```

You press ⟨DONE⟩ to leave Edit mode, then ⟨DONE⟩ again to leave Select mode. When you return to View mode, Cornerstone applies the selection criteria and displays only those records with a value of "Cambridge" for the *City* attribute.

Here's a second example that continues the previous scenario. Your working set contains just the Cambridge customers, but now you want to see the Cambridge customers who have ordered $2000 or more worth of material from your firm. You issue the command

```
SELECT BY-FORM
```

Cornerstone puts you back in Select mode, and displays the form in which you entered "Cambridge." To change the form, you issue the EDIT command, and continue pressing ⟨ENTER⟩ to move the cursor down to the *Total Purchases* field. Here you enter an expression that tells Cornerstone to look for values of 2000 or more in this field: the number 2000 followed an ellipsis (...).

```
TOTAL PURCHASES: 2000...
```

To select the records, you press ⟨DONE⟩ twice to return to View mode. Now the working set contains fewer records. To see that the total purchases are indeed from $2000 up, issue the command

```
COLUMN NAME CITY TOTAL_PURCHASES
```

After displaying the list of Cambridge customers with $2000 or more of total purchases, you can get back all the records in the file by issuing the command

```
SELECT ALL
```

The remainder of the chapter explains how to use all types of selection in Cornerstone, including details on selection by form, of which you have just seen two examples.

Selecting Flagged Records

When you want to work with a small number of records, and you can tell which ones you want just by looking at them, you can select them quickly by *flagging* them. To flag a record in View mode you move the record cursor to it then press the ⟨FLAG/UNFLAG⟩ key. The flag appears as the letter "F" to the left of the record. A flag is like a temporary check mark. It is not entered as data in a record, and it disappears when you issue the SELECT command or press ⟨DONE⟩ to leave View mode. You can flag as many or as few records as you want. To make your working set consist of the records you flagged, issue the command

```
SELECT FLAGGED
```

If you want to work with all but a few records in a file, you can flag the few records you *don't* want to work with and issue the command

```
SELECT UNFLAGGED
```

Using a Selection Form

You've seen how to select records using a selection form. A selection form for choosing September 1984's orders of $1000 or more looks like this:

```
ORDER DATE: 9/1/84 ... 9/30/84
TOTAL AFTER DISCOUNT: 1000 ...
```

To perform this kind of selection, issue the SELECT BY-FORM command from View mode. Now you're in Select mode, and the values you enter in the selection form determine which records will constitute the working set. You enter values in a selection form as you enter values in any other form: enter Edit mode and type them in. Each field in the selection form corresponds to an attribute in the file. Cornerstone compares the value or expression you enter in the field with the value for the corresponding attribute in each record. A successful match means the record is selected.

Properties of Selection Forms

Each *field* in a selection form corresponds to an *attribute* in a file, but they differ in two respects.

- □ You cannot directly enter a value in a record for a derived attribute, but you *can* enter a value in its corresponding field in the selection form. This means you can select records based on the value of a derived attribute.
- □ You can enter multiple values or expressions in a selection form even if its corresponding attribute can take only single values.

When you enter multiple values in a field in a selection form, Cornerstone selects those records whose value for the corresponding attribute matches at least one of those multiple values.

```
ORDER DATE: 9/1/84 ... 9/30/84
TERMS: COD
          NET30
TOTAL AFTER DISCOUNT: 200 ...
```

To be selected, a record must meet at least one of the selection criteria *in each field* in the selection form. In the example above, Cornerstone selects only the records which have an order date from 9/1/84 to 9/30/84, *and* have terms of either COD or NET30, *and* have a total purchase after discount of $200 or more.

When you return to View mode, the status line displays the number of records that meet the selection criteria. Sometimes the number is expressed as a range. (For example, the range "11...16" means that at least eleven but no more than sixteen records meet the selection criteria.)

☒ The range on the status line will become narrower as you scroll through the records or issue certain commands. To speed response time, Cornerstone checks to see whether a record meets the selection criteria only when it displays the record, or when you issue the SORT, REPORT TOTAL, or PRINT commands (which force Cornerstone to check all the records). So the range becomes a single number only when you issue one of these commands, or when all of the records have been displayed.

If no records in the file meet the selection criteria you entered, Cornerstone warns you of the fact, and no records are displayed. Perhaps the fact that no records meet the criteria tells you what you want to know. If not, you can issue the SELECT BY-FORM command and return to Select mode to edit the selection form as necessary. You'll notice that when you already have some selection criteria in effect, they are displayed when you return to Select mode; when you have no selection criteria in effect, Cornerstone displays a blank selection form for you to fill out.

☒ When you select by form, you always select from the *initial set*, not the working set. The initial set consists of the records you entered View mode with. Generally, your initial set is the entire file, but if you entered View mode with the ONLY-RELATED-RECORDS option, the initial set consists only of those related records.

☒ If you select by form and edit the selection form, the new criteria will select from the entire initial set, *not* just from the records that were selected by the original form.

To abort a selection in progress, press Ctrl-Break. You are then asked to enter YES to return to DOS, or NO to continue the selection from where you stopped it.

Selection Expressions

There are several special expressions that you can enter into a selection form. Each provides a different strategy for selecting records.

Exact match. If you know the exact value you want to match, you can just type that value in the appropriate field. (Upper and lower case are treated the same.) For example,

```
CITY: Cambridge
```

will select all records with the value "Cambridge" for the *City* attribute. You can use this kind of selection with values of any type.

Wild card. If you want to select records based on a partial match, or if you want to find a record, but you remember only part of a value, you can use an asterisk (*) to match any character or characters in a String.

For example,

```
NAME: Cor*
```

matches "Cornerstone," "cord," "coral," "corroborate"—any value that begins with "cor." You can put the asterisk at the beginning of the value:

```
STATE: *A
```

This expression matches "CA," "LA," "MA," "PA," "VA"—any state whose abbreviation ends in "a." You can also put an asterisk at each end of the value:

```
COMPANY: *ELECTRONICS*
```

This expression matches "Acme Electronics," "Electronics Emporium," "Bill's Electronics, Inc."—any value that has "electronics" anywhere in it. You can even put the asterisk in the middle of the value:

```
LAST NAME: *W*n*
```

This expression matches "Washington," "Wilson," "Newman"—any name that has a "w" somewhere before an "n."

You can use wild card matching for String values only.

Ranges. If you want to match values that are within a certain range, use an ellipsis (...). For example,

```
PRICE: 100...
```

selects records where *Price* is greater than or equal to 100.

```
ORDER DATE: ... 1/30/85
```

selects records whose order dates are on or before January 30, 1985.

```
LAST NAME: A ... L
```

selects records whose *Last Name* field has a name beginning with anything from "A" through "L."

The examples above are inclusive ranges, that is, they select the bounds of the range as well as anything in between. To make a range exclusive (so it does not select the bounds), just add the word EXCLUSIVE after the range expression. For example,

```
LAST NAME: A ... L EXCLUSIVE
```

selects records whose *Last Name* values begin with letters from "B" to "K."

The expression

```
SALARY: ... $29500 EXCLUSIVE
```

selects records whose *Salary* values are less than $29500.

If it's easier for you to think of open-ended ranges as "less than" and "greater than" or "less than or equal to" and "greater than or equal to," you can enter the abbreviated forms for those expressions: LT, GT, LE, and GE. Here are some pairs of equivalent expressions:

```
PRICE: LT 99.95
PRICE: ... 99.95 EXCLUSIVE

NUMBER OF ORDERS: LE 50
NUMBER OF ORDERS: ... 50

COMBINED SALARY: GT 50000
COMBINED SALARY: 50000 ... EXCLUSIVE

AGE: GE 65
AGE: 65 ...
```

In date and time fields, you can enter BEFORE and AFTER to designate open-ended ranges:

```
TIME SHIPPED: BEFORE 14:00
TIME SHIPPED: ... 14:00 EXCLUSIVE

ORDER DATE: AFTER MAY 11
ORDER DATE: MAY 11 ... EXCLUSIVE
```

You can use range expressions for String, Number, Date, and Time attributes.

Empty fields. To select records that have no value for a particular attribute, just enter the word EMPTY in that field:

```
CONTACTS: EMPTY
```

Non-empty fields. To select records that have *any* value for a particular attribute (that is, the attribute is not empty), use the asterisk (*) alone. For example,

```
CREDIT CARDS CARRIED: *
```

will select records that have at least one value for *Credit Cards Carried.*

NOT. If you want to select records that do *not* have a certain value or range of values, just type NOT in front of the expression:

```
CITY: NOT The Bronx
```

```
DUE DATE: NOT AFTER TODAY
```

Don't enter more than one NOT expression in the same field: they will effectively cancel each other out. Records not selected by the first expression would be selected by the second. For example, if you try to select only records with the name "Harry" by entering

(This is an example of what *not* to do.)

```
NAME: NOT Tom
      NOT Dick
```

the first expression will select Dick and Harry (because neither is Tom), and the second expression will select Tom and Harry (because neither is Dick). So the selected records are Tom, Dick, and Harry. You can solve this problem with the NEGATE command—see Negating a Selection Form, below.

Literal. Cornerstone interprets a "NOT" at the beginning of a String field to mean you want to select all records that do not have the value following the "NOT." But what if you want to select records in which a certain attribute has the value "Not applicable"? When you enter the selection expression

```
NAME OF SPOUSE: Not applicable
```

Cornerstone selects records where *Name of Spouse* has a value other than "applicable." To select records with the value "Not applicable," type a backslash (\) before the word "not."

```
NAME OF SPOUSE: \Not applicable
```

The backslash tells Cornerstone to treat the word that follows *literally,* instead of as a special selection term. Similarly, you can use the backslash in front of BEFORE, AFTER, GE, GT, LT, LE, '...', EXCLUSIVE, '*', EMPTY, and '\', all of which have special meaning in a selection form, as you have seen.

You can always press ⟨OPTIONS⟩ to see what kinds of selection expressions you can enter.

The CALCULATE Command

You can select records on the basis of more elaborate criteria by first creating a *derived* attribute with the CALCULATE command in Select mode.

Suppose you are working with the *Customer* file in the Sample Database and you want to narrow your working set to customers who purchase an average of $5000 or more per order. Since the total amount of each order is an attribute in the *Order* file (*Total After Discount*), you must derive the information from that file. You can create an attribute whose value is derived from the average (computed with the AVG function) of the total purchases. So, you would issue the CALCULATE command, which gives you a form in which you type the name of your new attribute and its derivation.

```
Name: AVG_TOTAL_PURCHASE
Derivation: AVG(ORDER.TOTAL_AFTER_DISCOUNT)
```

A new field will be added to the bottom of the form. Enter the range you want to select

```
AVG TOTAL PURCHASE: $5000 ...
```

Viewing Related Records

Another way to narrow your working set is to issue the VIEW command from View mode or Update mode. When you issue this command to view a related file, the system asks you whether you want to view *all* the records in that second file, or *only* the records that are related to a certain record or group of records in the first file. At that point you can narrow your working set by telling Cornerstone that you want to view, for example, records related to the flagged records.

Multiple Selection Forms

Many of your selection needs will be met using just one selection form. But you can add additional forms to your selection criteria for more elaborate selections. Let's say you want to print address labels for a special mailing to customers who have placed more than 10 orders, or who have purchased more than $1000 worth of merchandise. If you entered these criteria into a single form, you would get only records that satisfy *both* criteria.

To select records that satisfy one *or* the other criterion, enter the first criterion

```
NUMBER OF ORDERS: 10 ...
```

in the form, then press ⟨DONE⟩ and issue the OR-FORM command. Cornerstone displays a new form where you can enter the second criterion,

```
TOTAL PURCHASES: $1000 ...
```

The second form is called an OR-FORM. It searches the working set for records that meet its criteria, then adds these to the records selected by the previous form or forms. In our example, the OR-FORM looks for records with *Total*

Purchases of $1000 or more and adds these to the records selected by the first form, which selected records where *Number of Orders* was 10 or more.

Now let's modify our example slightly. Let's say we want to send the mailing only to customers in Boston. One way to do that is to add the criterion

```
CITY: Boston
```

to each of the forms already described. But a quicker way is to press ⟨LAST⟩ to move to the last form and issue the AND-FORM command. Like the OR-FORM command, it adds a new selection form to the set. But the AND-FORM tells Cornerstone to *narrow* the number of records that are selected. To be selected, a record must meet the criteria specified in the previous forms *and* the criteria in the AND-FORM. In our example, you would simply enter "Boston" in the AND-FORM to select customers who have placed 10 or more orders or have purchased $1000 worth of goods, *and* are located in Boston.

☒ When you issue the AND-FORM command, the new form is added immediately following the form that is currently displayed. Since an AND-FORM selects records that meet the criteria in both the AND-FORM and in the previous forms, the order of the selection forms is significant. You'll probably want an AND-FORM to follow all other forms, so make sure the last form is displayed when you issue the AND-FORM command. (Similar comments apply to OR-FORMs.)

If you want to delete a selection form from a set of selection criteria, issue the DELETE command in Select mode. You can use this command to delete either the current form or all the selection forms in a set.

When you have more than one selection form in a set of selection criteria, the Select mode status line tells you how many forms there are in the set, which one is currently displayed, and whether the form you see is an OR-FORM or an AND-FORM. You can move from form to form with the ⟨NEXT⟩, ⟨PREVIOUS⟩, ⟨FIRST⟩, and ⟨LAST⟩ keys.

Negating a Selection Form

Just as you can tell Cornerstone to select records which do *not* have a particular value (by typing NOT before the value), you can tell Cornerstone to select all records that would *not* be selected by the form by issuing the NEGATE command.

Returning to the example, suppose you have sent out your mailing to your Boston customers and now want to send it to the remaining customers. You issue the NEGATE command on the AND-FORM where you entered the criterion "Boston." When you press ⟨DONE⟩, Cornerstone selects customers *not* located in Boston.

The word NEGATE in the status line tells you that the form has been negated. To regain the original, unnegated form, you issue a second NEGATE command.

Saving Selection Criteria

When you type expressions in a selection form, you create a set of selection criteria. You can save a set of selection criteria for later use with the SAVE command. When you issue this command you enter a name for the set of selection criteria. For example, to save the selection criterion

```
TOTAL PURCHASES: ... $1000
```

under the name *Cheapskates,* you enter the command

```
SAVE SELECTION-CRITERIA CHEAPSKATES
```

You can issue the SAVE command in Select mode or in View mode. When you issue the command in View mode, you save the selection criteria currently in effect.

Once you have created and saved some selection criteria, you can use them later with the USE command in View mode or Select mode:

```
USE SELECTION-CRITERIA GOOD_CUSTOMERS
```

The effect of the USE SELECTION-CRITERIA command is the same as entering Select mode, filling out the form, and returning to View mode: Cornerstone selects the records that meet the criteria saved under that name.

☞ A set of selection criteria is tied to a particular file. ☜ You can USE a set of selection criteria on a file only if you saved it in that file.

When a set of selection criteria contains more than one form, the SAVE SELECTION-CRITERIA command saves all the forms under one name, and the USE SELECTION-CRITERIA command selects records on the basis of criteria in all the forms in the set you specify.

Once you save a set of selection criteria, subsequent changes are saved only if you issue the SAVE command again. You can save the edited selection criteria under the original name (in which case the original selection criteria are overwritten) or under a new name (if you want to keep the original selection criteria).

If you want to change a set of selection criteria permanently, you must issue the SAVE command again under the same name.

If you create some selection criteria and don't save them, Cornerstone will save them automatically under the name *Previous-Select.* Cornerstone saves only the last unsaved selection criteria, so you should get in the habit of saving them yourself.

You can delete any selection criteria you have saved by issuing the DELETE SELECTION-CRITERIA command in Initial mode. For example,

```
DELETE SELECTION-CRITERIA ORDER PAST_DUE
```

deletes the selection criteria called *Past_Due* that you created for the *Order* file.

The Selection Format

In Select mode you type selection criteria in a selection form. Like the forms in View mode and Update mode, your selection form can display as many or as few attributes as you need, and in any order and arrangement you like. When you enter Select mode, Cornerstone displays the selection criteria in the format you created and named *Select.* If you don't have a format called *Select,* Cornerstone displays the selection criteria using the DETAILED ALL-ATTRIBUTES format, which displays all the attributes, one per line.

You can also display selection criteria using any format you saved in View mode or Update mode. In Select mode, issue the USE command to recall a format. For example,

```
USE FORMAT INVOICE
```

displays your selection criteria using the format you named *Invoice.*

If a file doesn't have a format called *Select,* you can create one in Select mode with the DETAILED or FORMAT commands. The DETAILED command allows you to re-order the fields in the form, and display as few or as many fields as you need. You might want your selection form to display only the fields you plan to select on, or to display them in a special order. You might move a field to the top of the form if you plan to select by it frequently.

The FORMAT command lets you design more elaborate selection forms. This command brings you to Format mode, in which you can design your selection form to appear however you like. You can put more than one attribute on a line, or add constant text to the form. (You can learn more about Format mode in chapter 8, "Designing Forms.")

Whichever format you choose for your selection form, Cornerstone will select records based on the values and expressions you enter in the selection form. The format does not affect which records are selected.

☒ If you want to create a report using selection criteria and a report format that is sorted, or includes totals or subtotals, use the selection criteria first, then the report format. This prevents Cornerstone from having to compute the sorting, totals, or subtotals for the records that won't be selected.

☒ When you use a format that does not display a field in which you typed some selection criteria, those criteria are nevertheless in effect. Cornerstone will warn you if this should happen.

Printing the Selection Criteria

When you produce a report, you may want to print the selection criteria used to create the report. You can get a printout in Select mode with the PRINT command. This command lets you print one form or all the forms in the selection criteria.

ADVANCED REPORTING

6

CHAPTER

CHAPTER 6

CONTENTS

Advanced Reporting

Introduction

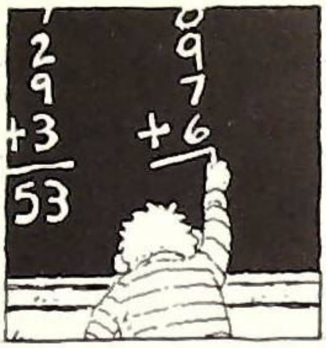
By now you should be familiar with the following basic reporting commands: COLUMN, DETAILED, SORT, SAVE, and USE. If you're not, consult chapter 3, "Viewing Your Data."

Your reporting needs will sometimes exceed the capabilities of these commands. You may want to include subtotals for different groups of records, or totals for the entire report, or both. Perhaps you want a report arranged in a way that's neither columnar nor row-oriented. Once you've learned the basics of Cornerstone, you can use advanced reporting features to produce just the report you want.

Cornerstone's interactive style also makes it easy for you to develop sophisticated reports gradually, one step at a time. This saves you time at the outset, because you don't have to plan out every last feature, and later on, because it's easy to make changes.

The REPORT Command

This chapter focuses on the REPORT command, found in the View mode menu. REPORT actually consists of several options that allow you to produce summaries (subtotals and totals), calculate derived attributes, and control the layout and design of the report. The layout and design option is only discussed briefly in this chapter; you will find a full discussion in chapter 8, "Designing Forms."



Using the REPORT Command

To use any of the reporting features, select REPORT from the View mode menu. A new menu appears, listing the command options, and you must select one. Remember: you can press ⟨HELP⟩ to get more information about the various options. The table below briefly describes each of the reporting options.

TOTAL. Performs summary operations on all records in the current working set.

SUBTOTAL. Performs summary operations on groups of records.

HIDE. Suppresses display of report elements (e.g., headers, subtotals, totals, records).

SHOW. Displays hidden report elements.

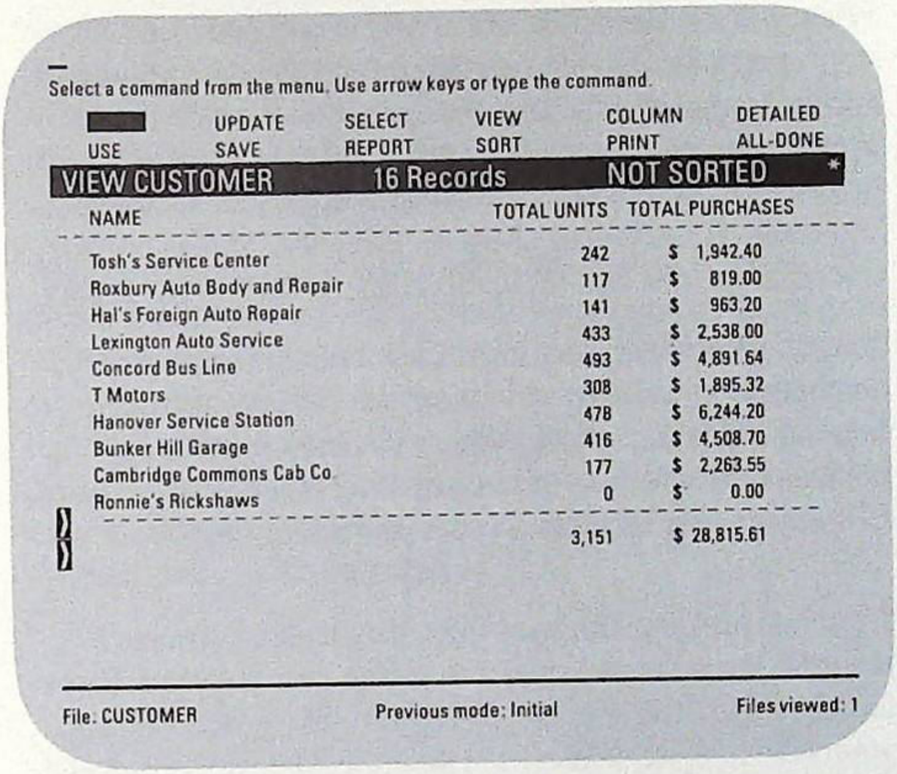
CALCULATE. Defines new derived attributes that can be used in the report.

FORMAT. Controls the layout and design of any report element.

Totals

Totals present summary information about all the records in the working set. For example, suppose you want to see the total number of units and total value of orders from all your customers. Using the *Customer* file from the Sample Database, enter the two commands

```
COLUMN NAME TOTAL_UNITS TOTAL_PURCHASES
REPORT TOTAL SUM TOTAL_UNITS
    SUM TOTAL_PURCHASES
```



```
_
Select a command from the menu. Use arrow keys or type the command.
              UPDATE      SELECT      VIEW        COLUMN      DETAILED
USE           SAVE        REPORT      SORT        PRINT       ALL-DONE
VIEW CUSTOMER             16 Records              NOT SORTED             *
NAME                                   TOTAL UNITS  TOTAL PURCHASES
-----------------------------------------------------------------------
Tosh's Service Center                          242   $  1,942.40
Roxbury Auto Body and Repair                   117   $    819.00
Hal's Foreign Auto Repair                      141   $    963.20
Lexington Auto Service                         433   $  2,538.00
Concord Bus Line                               493   $  4,891.64
T Motors                                       308   $  1,895.32
Hanover Service Station                        478   $  6,244.20
Bunker Hill Garage                             416   $  4,508.70
Cambridge Commons Cab Co.                      177   $  2,263.55
Ronnie's Rickshaws                               0   $      0.00
    ------------------------------------------------------------
                                             3,151   $ 28,815.61

File: CUSTOMER            Previous mode: Initial            Files viewed: 1
```

The accompanying figure displays the results of these commands. The first command creates a columnar report with the desired attributes. The second command creates the totals, which are displayed at the end of the report. This second command line may seem complex, so let's look at it piece by piece.

When you select REPORT TOTAL from the View mode menu, a new menu appears. This is a list of all the *aggregate functions* available with the TOTAL command. Cornerstone offers a wide range of choices, including (arithmetical) sum, (arithmetical) average, standard deviation, variance, minimum value, or maximum value. See the section below, Aggregate Functions.

When you use the TOTAL command, you choose an aggregate function and then choose the attribute you want it applied to. In the example, the aggregate function SUM is used twice, once for TOTAL_UNITS and once for TOTAL_PURCHASES. You can include as many aggregate functions as you want. When you've chosen all you need, press ⟨ENTER⟩. Cornerstone displays the totals at the bottom of the report.

You can recompute the totals at any time by issuing the command REPORT TOTAL without selecting an aggregate function. This is useful if you've added new records or performed a selection after computing totals, and you want to compute new totals for your new working set.

Subtotals

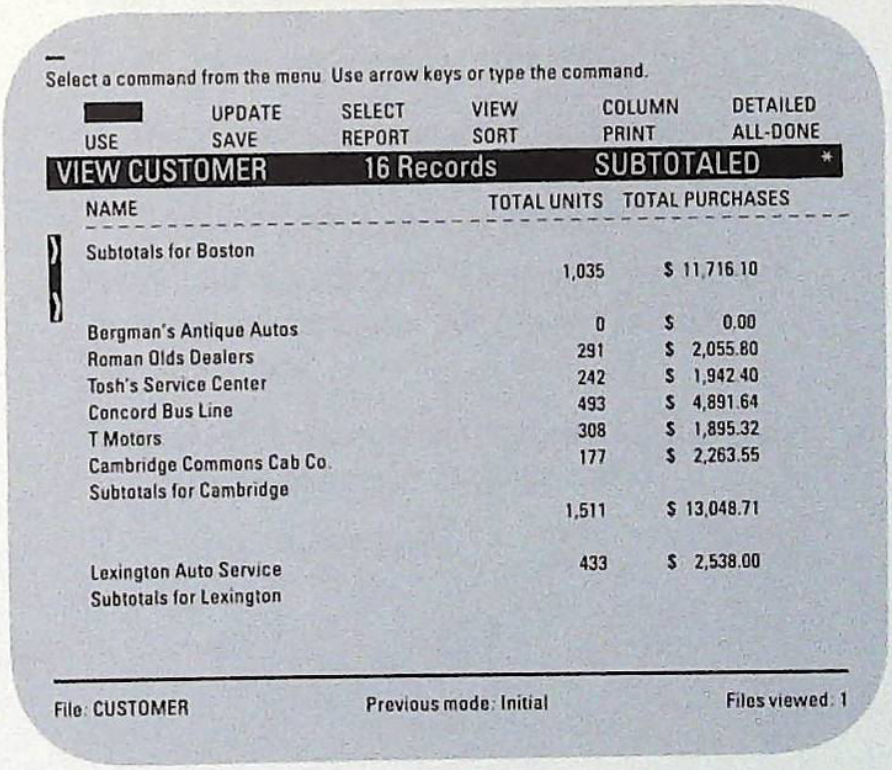
Subtotals present summary information about groups of records. For example, now that you've seen total purchases for all your customers, you might want to check how much was ordered by customers in each city. You could select just the records from a particular city and then compute the totals, repeating this operation for each city, but this would be cumbersome. Also, you probably want all the information to appear in the same report. Subtotals provide the solution.

Start with a report showing the relevant attributes. Subtotals or totals can be computed for any attribute, whether or not it's in the report format, but for purposes of clarity you will generally want to include any attribute that is being summarized. For the above example, start with the *Customer* file, and enter the commands

```
COLUMN NAME TOTAL_UNITS TOTAL_PURCHASES
REPORT SUBTOTAL SUM TOTAL_UNITS
   SUM TOTAL_PURCHASES
```

As you see, the REPORT SUBTOTAL command works much like the REPORT TOTAL command. You tell Cornerstone which aggregate function(s) to use (SUM in this case), and which attribute(s) to apply them to. After this point, though, the SUBTOTAL and TOTAL commands differ.

When you enter the subtotal command above, Cornerstone prompts you to select the attribute for grouping the subtotals. This is called the *grouping attribute*. In this case you want all records from the same city to be grouped together, so select CITY. (If your records aren't sorted by *City,* Cornerstone will ask you whether you want to sort before calculating the subtotals.) The figure below shows the new report with subtotals.



```
Select a command from the menu  Use arrow keys or type the command.
                 UPDATE       SELECT      VIEW        COLUMN      DETAILED
    USE          SAVE         REPORT      SORT        PRINT       ALL-DONE
VIEW CUSTOMER            16 Records               SUBTOTALED              *
    NAME                                       TOTAL UNITS   TOTAL PURCHASES
    --------------------------------------------------------------------------
    Subtotals for Boston
                                                    1,035    $ 11,716.10

    Bergman's Antique Autos                             0    $      0.00
    Roman Olds Dealers                                291    $  2,055.80
    Tosh's Service Center                             242    $  1,942.40
    Concord Bus Line                                  493    $  4,891.64
    T Motors                                          308    $  1,895.32
    Cambridge Commons Cab Co.                         177    $  2,263.55
    Subtotals for Cambridge
                                                    1,511    $ 13,048.71

                                                      433    $  2,538.00
    Lexington Auto Service
    Subtotals for Lexington

File: CUSTOMER                 Previous mode: Initial              Files viewed: 1
```

Perhaps you've noticed a similarity between sorting and subtotaling. ☞ Sorting orders the records on the basis of a particular attribute or attributes, grouping together those with the same value. Subtotaling computes summaries for groups of records with the same value for the grouping attribute. ☜

Since Cornerstone computes and displays subtotals each time the grouping attribute changes, you will generally want to sort before computing subtotals. In the example, if customers from different cities are interspersed, Cornerstone will compute subtotals each time the city changes.

As with REPORT TOTAL, if you issue the REPORT SUBTOTAL command without selecting an aggregate function, Cornerstone will recompute the subtotals.

Aggregate Functions

The table below describes the aggregate functions that you can use with the SUBTOTAL and TOTAL commands. Aggregate functions used with the TOTAL command are applied to all records in the working set. Aggregate functions used with SUBTOTAL are applied to each group of records having the same value for the grouping attribute.

SUM. Computes the sum of values in the group. For Number or Integer attributes only.

AVERAGE. Computes the average of values in the group. For Number or Integer attributes only.

MINIMUM. Computes the smallest value in the group.

MAXIMUM. Computes the largest value in the group.

FIRST. Takes the first value from the group.

LAST. Takes the last value from the group.

COUNT. Counts the number of records in the group. Unlike all the other functions, COUNT doesn't require an attribute since it just counts the records.

STANDARD DEVIATION. Computes the standard deviation for the group. For Number or Integer attributes only.

VARIANCE. Computes the variance for the group. For Number or Integer attributes only.

NET PRESENT VALUE. Computes the net present value for the group. For Number or Integer attributes only. This function requires you to supply an interest rate. Cornerstone will assume that the records are consecutive time periods, and that the first record is time zero. Be sure to set the interest rate appropriately (12% per annum is .12 if each record represents a year, .01 if each record represents a month). Select only those records representing the time periods you're analyzing, and sort by date.

What's a Report?

You've now encountered all the elements of a Cornerstone report: record data, header (created automatically by the COLUMN command), subtotals, and totals. Cornerstone considers each of these as a separate *report element.* A report can contain any or all of these elements.

Whichever elements you include in the report, Cornerstone saves them all together with SAVE FORMAT, and restores them all with USE FORMAT. Two other pieces of information are also saved with the report format: sorting specifications (see chapter 3, "Viewing Your Data"), and printing characteristics (see chapter 7, "Printing").

Hiding Report Elements

Cornerstone allows you to work with each report element individually. You can modify the appearance of any report element: For example, you can change the location and width of fields, add labels and constant text, or suppress field names. For details on using Format mode to alter the appearance of a form, see chapter 8, "Designing Forms."

You can also suppress the display of individual report elements. For example, suppose you will be going over last year's sales figures with the board of directors, and want to show them only the subtotals for each city and the overall totals. (They're not interested in the gory details.) In this case, you would want to hide the record data. Starting with the earlier report, just select

```
REPORT HIDE RECORDS
```

HIDE does not affect the report element—it only removes it from the display.

The SHOW command redisplays previously hidden report elements. To show the record data, just enter

```
REPORT SHOW RECORDS
```

You can even save a report with hidden elements and display them later when you use the report.

Creating Derived Attributes

The CALCULATE command lets you compute new data on the basis of existing data in your files. Working with the *Order* file, you might want to compute the average unit price for each order. Just enter

```
REPORT CALCULATE NEW-FORMULA
```

Cornerstone then asks you for a name and a derivation (formula). In this case, you could enter

```
Attribute Name: Average_Unit_Price
Derivation: Total_After_Discount / Total_Qty
```

Using REPORT CALCULATE is *identical* to defining a derived attribute in Define mode. This command saves you time by allowing you to define an attribute without leaving View mode.

Any new attribute you create with REPORT CALCULATE is appended to each record, and is permanently added to your file. You can delete it in Define mode, as you would any other attribute.

You can also use REPORT CALCULATE to explore and test out predictions. With the REPORT CALCULATE OLD-FORMULA option you can change the derivation of an attribute, examine the results, and then change it back, if necessary.

For more detailed information on derivation expressions and the use of derived attributes, see chapters 9 and 10, "Planning Your Database" and "Defining Your Database." Also refer to "Entering Information Into Forms" in *Owner's Handbook II*.

Designing Custom Report Formats

The REPORT FORMAT command takes you to Format mode, where you can freely modify the appearance of any of the report elements (record, header, subtotal, total). In Format mode you can modify the appearance of the report element as a whole (such as where the fields appear) or of the individual fields (such as the width of the field display). Chapter 8, "Designing Forms," describes Format mode.

Putting It All Together

The reporting process involves four basic steps:

□ View the file you plan to work with.

□ Select the records you want. (This may be all the records in the file, or only those records meeting specific criteria. See chapter 5, "Selecting Records," for details.)

□ Design the report format (appearance). This includes choosing which report elements you want (records, header, subtotals, total), which attributes should be displayed, and how each element should appear.

□ Print the report, if necessary.

With Cornerstone, steps 2 and 3 are separate and can be performed in either order. You could select just the record for Roxbury Auto Parts, design a report format (even including subtotals and totals), and then select all the records. Or you could design a report format and then eliminate unnecessary records. It's generally better to select the appropriate records first and then use the desired format, since you will have to re-sort or recompute subtotals after selecting records.

PRINTING

7 CHAPTER

CHAPTER 7

CONTENTS

Printing

Introduction

This chapter discusses the various ways of printing information with Cornerstone. The major use of printing is, of course, to produce printed reports. But Cornerstone can also print a variety of other information, including file definitions, newly updated records, selection forms, and empty data entry forms.

Printing Reports

Although one hears talk of the "paperless office of the future," most computer applications, and most database applications, still involve printing. You might, for example, want to print periodic reports (such as monthly sales figures), mailing lists, order forms, or form letters. All can be easily printed using Cornerstone.

The term *printing a report* means printing data from a set of records in View mode. The report can include one or more report elements (records, header, totals, or subtotals).

The basics are simple. Select the PRINT command from the View mode menu. PRINT PRINTER prints on your printer; PRINT DISK writes to a disk file.

Cornerstone has many built-in defaults for printing your report (for example, spacing, page size, and margins). You can modify these as you like.

Cornerstone also has information about a variety of printers. You can add information about your printer, if necessary. Both the report layout and printer options can be modified with the PRINT SETUP command.

Editing the Print Form

When you choose PRINT SETUP you are placed in Print mode with a new set of commands and a form to edit, called the *print form*. This form contains quite a few options (it extends beyond the first screen), which determine the appearance of your report. To edit the form, just press ⟨ENTER⟩ to issue the EDIT command.

☞

The following is a brief description of each printing option. Most are self-explanatory. ☞ Remember, you can always press the ⟨HELP⟩ key for advice. ☜

Number of copies. Sets the number of copies to be printed.

Left margin. Sets the number of spaces for the left margin. Notice that there is no right margin option. Your report format determines the width of the report (up to 255 characters per line).

Top margin. Sets the number of lines for the top margin.

Bottom margin. Sets the number of lines for the bottom margin.

Number of lines per page. Sets the page length. Most printers use 6 lines/inch, therefore the default is set at 66 for 8½″ × 11″ paper. You can use this option in conjunction with *Maximum number of records per page* to generate mailing labels. See the section below, Tips on Printing.

Print header on every page? Answer "Yes" or "No."

Print title on separate page? Answer "Yes" or "No."

Keep records together? Answer "Yes" or "No." Answering "Yes" will prevent Cornerstone from splitting a record on to two pages, where possible.

Maximum number of records per page. Sets the maximum number of records per page. Entering "1" starts each record on a new page.

Number of blank lines between records. Sets the number of blank lines after each record. (You can also insert blank lines into your record form itself, using REPORT FORMAT. This often improves the legibility of the report.)

Start new page for subtotal group? Answer "Yes" or "No." Answering "Yes" starts a new page after each set of subtotals.

Extra lines between subtotals at top. Inserts the specified number of blank lines between a group of records and its subtotal information.

Extra lines between subtotals at bottom. Inserts the specified number of blank lines between subtotals and the next group of records.

Page number style. Determines the position of the page number. Options are "none" (no page number), "top," or "bottom."

Print date on top of page? Answer "Yes" or "No." Answering "Yes" prints the current date at the top of every page in the report.

Use pre-printed form? Answer "Yes" or "No." Answering "Yes" sets a number of characteristics for using pre-printed forms. See the section below, Tips on Printing, for details.

Stop every page? Answer "Yes" or "No." Answering "Yes" interrupts printing after each page, allowing you to feed individual sheets of paper.

Title style. Specifies the style of typeface for printing the title. Options are plain, bold, underline, italics, shadow, and special (defined by you). Any choice must be supported by your printer.

Header style. Same as title style, for printing the header.

Record style. Same as title style, for printing the records.

Subtotal style. Same as title style, for printing the subtotals.

Total style. Same as title style, for printing the totals.

Once you're in Print mode (via the PRINT SETUP command) you can print your report in either of two ways: by selecting PRINT from the Print mode menu or by returning to View mode (by pressing ⟨DONE⟩), and selecting PRINT from that menu.

Cornerstone saves whatever printing instructions you establish when you save the report format in View mode. If you wish to save the instructions (as you will if you've gone to the trouble of changing them), press ⟨DONE⟩ after editing the print form, then select SAVE FORMAT.

The Print Mode Menu

This section describes the other commands in the Print mode menu: DEVICE, FORMAT, SHOW, and PRINT.

Devices

The DEVICE command is used to supply information to Cornerstone about your printer, such as how it underlines, italicizes, and issues a carriage return. In addition, DEVICE can be used to select a printer if you have more than one attached to your system, or to specify the style of printing (draft or letter-quality).

A device is a set of printer instructions, and each device has its own name. You can specify which device (which set of printer instructions) to use whenever you print. If you don't choose a device, Cornerstone will use its default.

The DEVICE command has two options: CREATE and ALTER. DEVICE CREATE is used to enter information that will be stored under a new device name. DEVICE ALTER is used to select or change the information about an existing device.

Pre-defined devices for several printers have been supplied with Cornerstone. DEVICE ALTER will display the list. If one of these is your printer, complete the command by selecting the appropriate name. This will establish that printer as your *current device,* which is the device automatically chosen by Cornerstone for all printing operations. You can check the current device name by referring to the Print mode status line.

Even if your printer isn't listed, one of the existing devices may meet your needs. Before defining a new printing device, try selecting one of the pre-defined devices. If this doesn't work well, consult your printer manual for specific information.

Selecting either DEVICE command puts you in Edit mode, editing the device form. The various elements of this form are described below:

Name. This is the name for the device. In most cases you'll just want to use the name of your printer.

Physical device. This specifies the physical location of the output device. The options are LPT1 (use this if your printer is using the normal printer port) or LPT2.

Initialization string. This specifies an initialization string to be sent to the printer when you start printing. This is useful if you need to initialize the printer or select options.

Reset string. This specifies a reset string to be sent to the printer when you are done printing. This is useful if you need to reset any options you set up with the *Initialization string* above.

Line feed string. This specifies the hexadecimal characters that the device interprets as a line feed. The standard value is carriage return followed by line feed, which is 0D0A in hexadecimal.

Print style. This section of the device form specifies the hexadecimal strings for turning on or turning off various print style instructions such as boldface, underline, italics, shadow, and special.

Form feed string. This specifies the hexadecimal characters that the device interprets as a form feed. The standard value is hexadecimal 0C.

If you want to delete unnecessary devices (you'll only need the ones for your printers), select DELETE DEVICE from the Initial mode menu.

Titles

The FORMAT command from the Print mode menu is used to create or change a title for your report. The SHOW command is used to display the title. Titles can be constant text, for example "MONTHLY REPORT," or they can contain summary information computed from records in your report.

☒ For example, you could insert two summary fields, MIN ORDER_DATE and MAX ORDER_DATE, into your title form, and have the title: "SALES FROM ________ TO ________". For more information on this use, see chapter 8, "Designing Forms."

Print

The PRINT command can be used to "print" to a disk file, as well as to your printer. If you select the PRINT DISK command, Cornerstone creates a text file on your current data disk. The PRINT DISK command requires you to enter a name for the text file. You can simply enter a file name or you may enter an entire file specification, with a drive designator, path name, file name, and extension. The name you enter is subject to the usual restrictions on DOS file specifications (listed in your DOS manual).

Tips on Printing

Mailing Labels

To produce mailing labels start with a row-style format containing the appropriate fields. Use the SET-CHARAC command in Format mode to suppress display of the field names (you don't need "Name:" before each person's name). You can establish the appropriate label size in two different ways.

□ You can make the form the same number of lines as the label (adding blank lines if necessary); or

□ You can set the page size in the Print form equal to the number of lines in the label, and set the maximum number of records per page to "1."

The first approach is easier if your addresses do not contain multi-valued attributes (such as street address). The second approach is safer if you do have multi-valued fields.

Cornerstone will print only one column of labels per page.

☒ One easy way to display the city, state, and zip on the same line is to define a new attribute that combines these three attributes, using the STRING function. For example, create a derived attribute called *City_State_Zip*, and enter the derivation expression: STRING*(City,", ",State,"   ",Zip)*. This puts a comma followed by a space between the city and state, and two spaces between the state and zip.

Pre-printed Forms

Cornerstone has some special features to let you conveniently insert data into the correct place on pre-printed forms. If you answer "Yes" to the question *Use pre-printed form?* Cornerstone makes two changes to its normal handling of forms:

First, it suppresses display of all field names. (If you need to add a field name to your form you can always use constant text in Format mode.)

Second, Cornerstone prints each field or constant text starting on the *exact* line specified in your format. How does this differ from the normal case? Normally Cornerstone inserts as many lines as are necessary to display the information from your records. For example, if your form includes a multi-valued field or subrecord, Cornerstone will make additional room for each value. Cornerstone will also insert extra lines for any single-valued attribute whose value is longer than the specified field length. In this sense the length of the normal Cornerstone form is not fixed, but changes with the length of the data.

When you choose the pre-printed form option Cornerstone prints each field (or constant text) starting on the exact line you specify in your format. Values will be extended, if necessary, into subsequent blank lines, but are terminated at the next line containing another field or constant text.

☒ Sometimes a blank line in your Cornerstone format corresponds to a pre-printed or otherwise reserved area in your existing form ("Don't write in shaded area" or "For office use only"). To prevent extension of a field into such an area, you must put some constant text on that line in the Cornerstone format. A period or dash in the corner will suffice.

Chapter 8, "Designing Forms," tells you how to set up your data and design Cornerstone forms for printing on your pre-printed forms.

Form Letters

You can even print form letters with Cornerstone. The repeating part of the form letter could be either constant text (entered in Format mode) or a string attribute (or attributes) that you copy from record to record with the COPY command (CTRL-D). Using attributes allows you to make editing changes in individual letters, which can be useful if there are slight changes from one letter to another. Fields containing the variable information (such as name or address) can be placed wherever necessary in the form.

You may find it more convenient to use a separate word processing system to create and process your form letters. You can still use Cornerstone to store names and addresses. The chapter on the Convert utility in *Owner's Handbook II* describes how to select Cornerstone data and merge the data with other word processing and form-letter programs.

Interrupting Printing

If you start printing a report and want to terminate printing prior to completion, just press ⟨CANCEL⟩ or Ctrl-Break.

Printing in Other Modes

The PRINT command appears in other modes besides View. In each it performs a specialized function. More information about the following uses can be found in the corresponding chapters.

Define. The PRINT command in Define mode prints the file and attribute definitions.

Update. The PRINT command in Update mode prints either the record currently being updated or all records in the update set.

Select. The PRINT command in Select mode prints either the current selection form or all selection forms in the current selection criteria.

Format. The PRINT command in Format mode prints either a blank copy of the form (one without data) or a grid that marks off horizontal and vertical spacing. A grid is useful when designing your Cornerstone forms to match existing pre-printed forms.

DESIGNING FORMS

8

CHAPTER

CHAPTER 8

CONTENTS

Designing Forms

Introduction

As you become a more demanding Cornerstone user, you will want to create your own custom reports. You may decide that you don't like a field in a particular place and would rather move it somewhere else, or that you want to draw a line under a particular field. This chapter introduces Format mode, which lets you do this sort of thing and more.

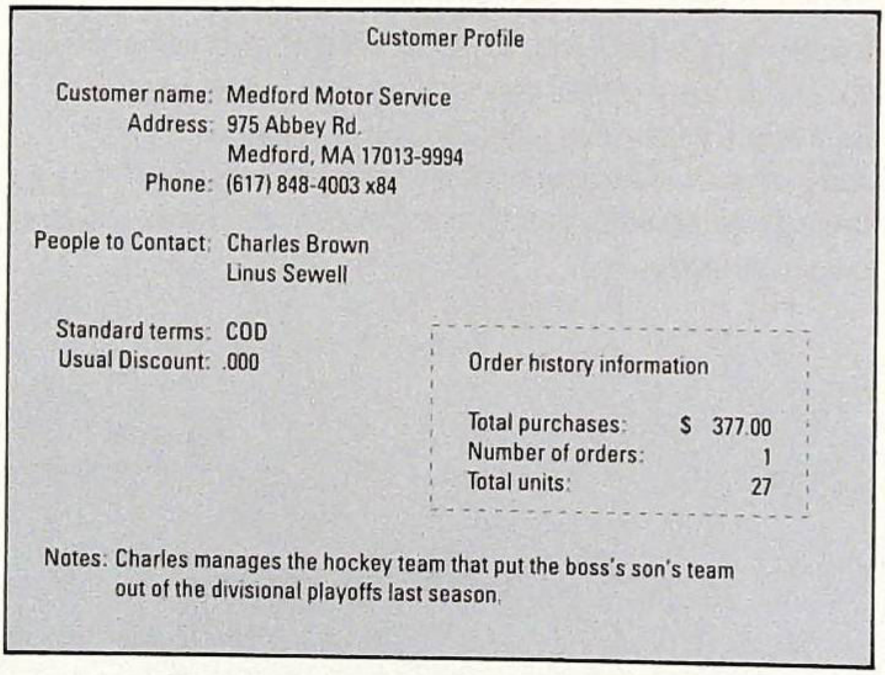
Although Format mode gives you complete control over the appearance of your data, Cornerstone is designed so you can handle most situations without invoking Format mode. The COLUMN and DETAILED commands provide an adequate solution for most reporting needs. It's when you want a special report, one that must look *perfect,* that you need the flexibility Format mode provides.

Format mode is fun to use once you become familiar with it. You can experiment with different designs, and when you've arrived at one you like, you can save it for future use. You need never worry that anything you do might harm your data—it can't. The figure below contrasts a detailed form (created using the DETAILED command) with a form designed in Format mode.

Form created using the DETAILED command

NAME: Medford Motor Service
STREET: 975 Abbey Rd.
CITY: Medford
STATE: MA
ZIP: 17013-9994
AREA CODE: 617
PHONE: 848-4003
EXT: 84
CONTACTS: Charles Brown
Linus Sewell
TERMS: COD
DISCOUNT: .000
NOTES: Charles manages the hockey team that put the boss's son's team out of the divisional playoffs last season.
TOTAL PURCHASES: $ 377.00
NUMBER OF ORDERS: 1
TOTAL UNITS: 27

User-designed format containing the same information


Customer Profile

Customer name: Medford Motor Service
Address: 975 Abbey Rd.
Medford, MA 17013-9994
Phone: (617) 848-4003 x84

People to Contact: Charles Brown
Linus Sewell

Standard terms: COD
Usual Discount: .000

Order history information

Total purchases: $ 377.00
Number of orders: 1
Total units: 27

Notes: Charles manages the hockey team that put the boss's son's team out of the divisional playoffs last season.

Overview and Concepts

You can get to Format mode from several different places in Cornerstone. Which mode you come from determines the form(s) you can format.

From View mode: The REPORT FORMAT command lets you format any element of the report: records, subtotals, total, or header. You select the form you want to format as part of the command. (You can even format two forms at once—see the section below, Working with Two Forms.)

From Update mode: The FORMAT command in Update mode lets you format the update form, which you use to create or change records.

From Select mode: The FORMAT command in Select mode lets you format the selection form, which you use to enter criteria for selecting records.

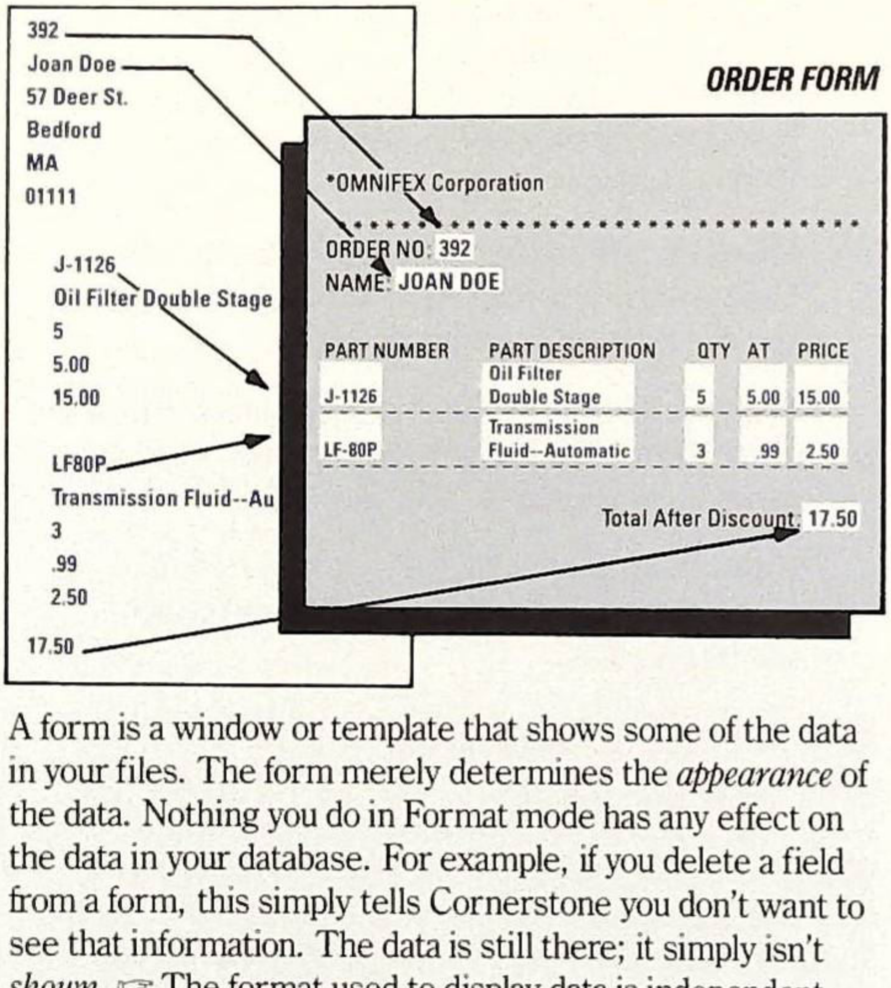
From Print mode: The FORMAT command lets you format the report title, which can be used when printing reports.

As you can see, the command you use to get to Format mode is generally FORMAT. The only exception occurs in View mode, where the command is REPORT FORMAT. Whichever mode you come from, once you're in Format mode, designing the form works pretty much the same way. There are some minor differences when you're working with subtotal, total, header, and title forms.

What's a Form?

You use forms throughout Cornerstone to supply information that would be inconvenient to enter on a command line. For example, you fill out a form whenever you create a new record or supply Cornerstone with information—such as when you customize the system or define a new attribute within a file.

ORDER RECORD



ORDER FORM

392
Joan Doe
57 Deer St.
Bedford
MA
01111

J-1126
Oil Filter Double Stage
5
5.00
15.00

LF80P
Transmission Fluid--Au
3
.99
2.50

17.50

*OMNIFEX Corporation

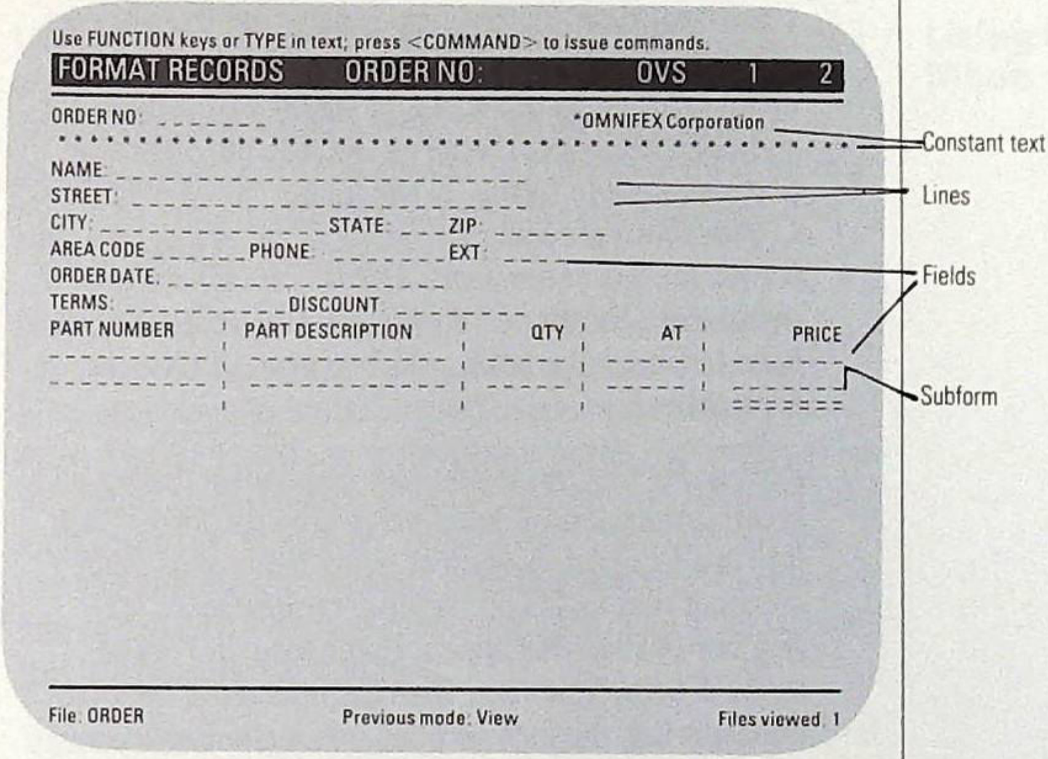
ORDER NO: 392
NAME: JOAN DOE

PART NUMBER	PART DESCRIPTION	QTY	AT	PRICE
J-1126	Oil Filter Double Stage	5	5.00	15.00
LF-80P	Transmission Fluid--Automatic	3	.99	2.50

Total After Discount: 17.50

A form is a window or template that shows some of the data in your files. The form merely determines the *appearance* of the data. Nothing you do in Format mode has any effect on the data in your database. For example, if you delete a field from a form, this simply tells Cornerstone you don't want to see that information. The data is still there; it simply isn't *shown.* ☞ The format used to display data is independent of the data itself. ☜

In Format mode, forms are displayed without any data in the fields. Instead, underbars appear where you would enter data. This lets you see how wide the field is. All forms contain the same basic elements.



```
Use FUNCTION keys or TYPE in text; press <COMMAND> to issue commands.
FORMAT RECORDS      ORDER NO:                    OVS     1     2
ORDER NO: -------                         *OMNIFEX Corporation
.....................................................................
NAME: -------------------------
STREET: ------------------------
CITY: --------------STATE: ---ZIP: -------
AREA CODE ------PHONE: -------EXT: ---
ORDER DATE: ----------------
TERMS: ----------DISCOUNT: --------
PART NUMBER  | PART DESCRIPTION  |   QTY |   AT  |    PRICE
----------   | ------------      | ----- | ----- | -------
----------   | ------------      | ----- | ----- | -------
             |                   |       |       | =======

File: ORDER            Previous mode: View            Files viewed: 1
```

Fields. Fields are holes or slots used to hold data. You enter data where the underbars appear in Format mode. For records, each field corresponds to an attribute in your file. For example, the *Name* field in the format in the figure above corresponds to the *Name* attribute in the *Order* file. When you use commands such as COLUMN and DETAILED, Cornerstone uses the names of the attributes you select as field names, but there's no rule that says the field name must be the same as the attribute name. You can change the name of a field to be whatever you like. (You'll learn how to do this later on in this chapter.) ☞ With Cornerstone, fields and attributes are not the same thing. Attributes belongs to *files;* fields belong to *forms*. A field is simply a place to display or print the value of an attribute. ☜

Constant text. Forms can contain text that appears whenever the form is displayed or printed. The text might give the form a name or supply instructions for someone filling out the form. You can put whatever text you want in forms. In the figure, "OMNIFEX Corporation" is constant text, as is the row of asterisks across the top of the form. The text is called *constant* because it's the same for every record, unlike the data used to fill out the form, which can be different for each record.

Lines. Format mode provides commands that let you create, move, and delete lines within the form. Each line in a form can contain as many fields or as much constant text as fits on the line, or none at all. The number of lines that appear in the form in Format mode is sometimes less than the number of lines that appear when the form is filled out, since Cornerstone creates extra overflow lines when a value is too long to fit in its field.

Subforms. A form may contain subforms corresponding to subrecord attributes (such as *Line Items*) in the file. The fields that appear in the subform correspond to (sub)attributes in the subfile. A subform is like a form within a form (just as a subrecord is like a record within a record). Cornerstone uses the subform as a template to display each subrecord.

You needn't worry about the size of the form unless you are printing reports that must go on a pre-printed form, since Cornerstone automatically handles subrecords, multi-valued fields, and values that overflow.

Using Format Mode

When you first enter Format mode, the cursor is positioned on the first field in the form. You can move the cursor around the form with the arrow keys, or you can use the motion keys (such as ⟨NEXT⟩ and ⟨PREVIOUS⟩) to move among the fields in the form. You can also use any of the scrolling keys to scroll the display.

In Format mode, if the cursor doesn't lie within a field, anything you type simply goes into the form as constant text. For example, if you wanted to create a title for your form, you could just move the cursor to the top and type the title. You can also use most of the editing keys used in Edit mode, such as ⟨RUB-CHAR⟩, ⟨RUB-WORD⟩, and ⟨DELETE⟩. You can also use the ⟨INS/OVS⟩ key to change the type-in style, just as in Edit mode. You cannot type text into a field, since this is where data is displayed.

You can issue commands in Format mode, but you must first press the ⟨COMMAND⟩ key. Otherwise, Cornerstone will treat what you type as constant text. When you press ⟨COMMAND⟩, Cornerstone displays a menu of commands, and you select the command you want in the usual fashion. For convenience, many of the commands in Format mode are assigned to control keys, so you only have to press the key to execute the command. Your Quick Reference Card lists the commands and the keys they are assigned to.

Commands

This section contains descriptions of all the commands in Format mode. You may want to skim it first just to get an idea of what's available. You can consult it again later if you need more information about a particular command.

The number of commands may seem large at first, but you'll find that you use some more than others, and that you can accomplish most tasks using just a few commands.

Manipulating Fields

The following commands work with a field in the form. To use these commands, first move the cursor to the field, then press the ⟨COMMAND⟩ key and issue the command.

INSERT FIELD

The INSERT FIELD command lets you add a new field to the form. When you issue the command, you must select an attribute from the menu. This is the attribute whose value is to fill the field. The menu shows only those attributes that don't already appear in the form. (When working with subtotal, total, header, or title forms, this command works a little differently—see the section below.)

After you issue the command, Cornerstone prompts you to move the cursor to the place you want the field to appear. Move the cursor to the desired location and press ⟨ENTER⟩. The new field is inserted at the cursor location.

MOVE FIELD

The MOVE FIELD command lets you move a field to another place in the form. It has three options.

> **NAME-ONLY** moves just the field name.
>
> **FIELD-ONLY** moves just the field itself (i.e., the underbars).
>
> **BOTH** moves the entire field, and is the default option.

Whichever option you select, the command works this way: Cornerstone prompts you to use the arrow keys to move the field. The field moves around the form as you move the cursor. When you are satisfied with the position, press the

⟨ENTER⟩ key to complete the command. You can press ⟨CANCEL⟩ at any time to cancel the command and send the field back to its original position.

☒ For both the INSERT FIELD and MOVE FIELD commands, Cornerstone will do its best to fit the field where you want it (it will even shrink the field width or suppress the field name if necessary), but if there's no room, the field will go on a new line under the cursor. Sometimes Cornerstone will shrink the width to a scant one character in addition to suppressing the field name. When this happens, the display can be confusing, especially if the field is adjacent to other fields in the form. If it appears that no field was inserted, look again! You can always press ⟨FIRST⟩ and then tab through the fields using ⟨NEXT⟩ to see where all the fields are.

☒ There is only one restriction that Cornerstone imposes when you move a field (or line): you may not move a field into or out of a subform. Cornerstone will display an appropriate error message if you attempt to do so.

DELETE FIELD

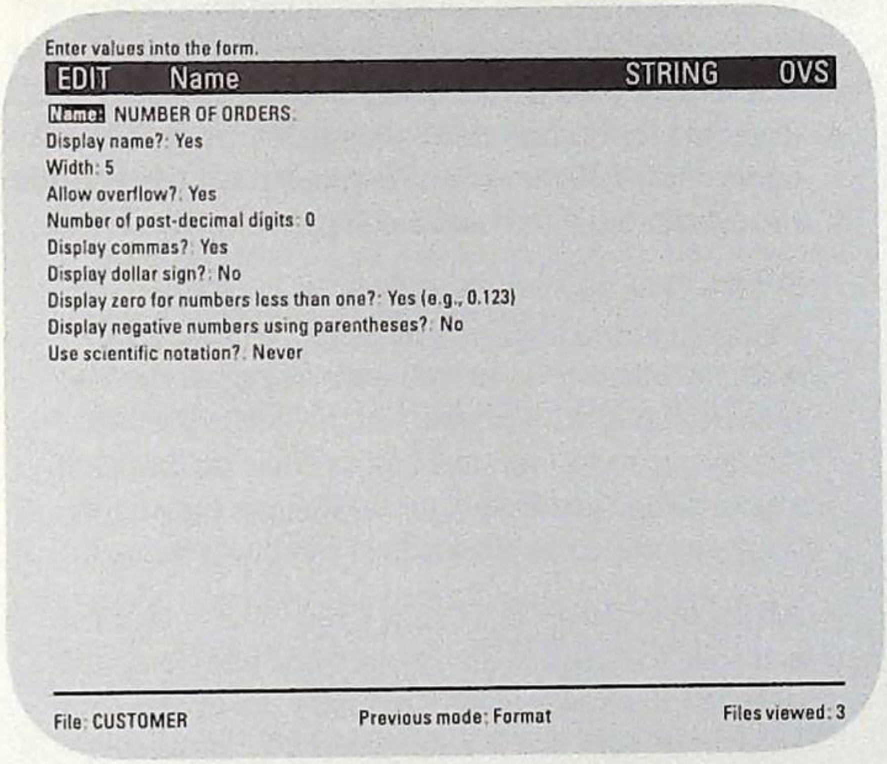
The DELETE FIELD command lets you remove a field from the form. To use it, move the cursor to the field you want to delete and issue the command. Deleting a field from the form has no effect on your database. When you delete a field all you're doing is telling Cornerstone not to display that field in the form. You can always INSERT it again later, if necessary.

Setting Field Characteristics

The commands we've described so far affect the position of individual fields in the form. You can also change the display characteristics of the fields themselves.

SET-CHARAC

The SET-CHARAC command lets you set various characteristics of the field, such as its name and width. The cursor must be in a field when you issue this command.



```
Enter values into the form.
 EDIT      Name                                        STRING      OVS
Name: NUMBER OF ORDERS.
Display name?: Yes
Width: 5
Allow overflow?: Yes
Number of post-decimal digits: 0
Display commas?: Yes
Display dollar sign?: No
Display zero for numbers less than one?: Yes (e.g., 0.123)
Display negative numbers using parentheses?: No
Use scientific notation?: Never

File: CUSTOMER               Previous mode: Format               Files viewed: 3
```

When you issue this command, Cornerstone gives you a form to fill out. The fields in this *Cornerstone* form let you specify the characteristics of the field in the form you are designing. The top line of the form shows the name of the attribute (or aggregate expression, in the case of a summary form) whose value is displayed in the field. The figure above shows a sample *field characteristics form.* Different fields have different characteristics depending on the type of data that goes in the field, but all fields have certain basic characteristics.

> **Name.** Every field has a name. Whenever Cornerstone creates a field, it selects a name for the field, typically the name of the attribute the field gets its value from. You can change the name if you wish. Just enter the name you want in the form.

Display name? You can tell Cornerstone not to display the field name —just answer "No" here. The field always *has* a name, though, even if it's not displayed. In Format mode you can always find out the name of a field (even when it's not displayed) by moving the cursor to the field and looking at the status line.

Width. The width of the field is the number of characters used to display its value. You can set the width to whatever value you want as long as there's room in the form. The width of the field is used for display purposes only, and has no effect on the actual size of data in your file. You can also use the ADJUST-WIDTH command to set the field width (see below).

Allow overflow? Normally, if the value in your file is too big to fit in a field, Cornerstone will create extra overflow lines as required to display the excess data. You can prevent this from happening by answering "No" here. If you answer "No," any extra data that doesn't fit in the field won't appear.

Editable? You can prevent anyone from entering or changing the data in the field by answering "No" to this question. This characteristic is useful when you intend to use the form to update records. For example, you might design a special update form for data entry personnel in which they cannot edit certain fields. Not all fields have the *Editable?* characteristic. For example, since derived fields are never editable, Cornerstone doesn't ask.

The other field characteristics all relate to display styles and are self-explanatory. For example, *Display dollar sign?* tells Cornerstone whether or not to display a dollar sign with the value of the field. These characteristics are the same as the ones used when you create an attribute in Define mode.

☒ When you set a display characteristic for a particular field in a form, the value you set overrides the value you set for the attribute, but just in the current form. For example, if you answered "No" to *Display commas?* when you defined the attribute, but answered "Yes" when you set the field characteristics for the field, then commas will be displayed whenever you use that form. Commas will not be displayed in other forms (unless you set the field characteristics in the other forms).

ADJUST-WIDTH

Once you start experimenting with Format mode, you may find yourself changing the widths of fields frequently. For this reason, Cornerstone provides an ADJUST-WIDTH command that lets you set the width of a field more conveniently than with SET-CHARAC.

To adjust the width of a field, just move the cursor into the field and issue the ADJUST-WIDTH command. Cornerstone then prompts you to use the ⟨ARROW-RIGHT⟩ and ⟨ARROW-LEFT⟩ keys to increase or decrease the width. This way you can see how wide the field is without bothering to figure out the actual width. Press ⟨ENTER⟩ to complete the command or ⟨CANCEL⟩ to restore the original width.

Manipulating Lines

The commands INSERT, DELETE, and MOVE also work with lines.

INSERT LINES

The INSERT LINES command lets you add blank lines to a form. First, move the cursor to the spot where you want to insert lines, then issue the command. You have the option of inserting the lines BEFORE or AFTER the line the cursor is on. When you issue the command, Cornerstone inserts a blank line, and prompts you to press ⟨ARROW-DOWN⟩ if you want to add more. The lines are highlighted as you add them. Press ⟨ENTER⟩ to complete the command. If you press ⟨CANCEL⟩, the command is canceled and no lines are inserted.

☒ When the cursor lies within a subform, lines are inserted into the subform, except when the subform lies at the end of the form. In this case, lines inserted AFTER the last line in the subform are inserted into the form, not the subform.

MOVE LINE

The MOVE LINE command lets you move a line in a form, including any fields or constant text it may contain. To move a line, go to the line and issue the command, then move the cursor to the new location and press ⟨ENTER⟩. You can press ⟨CANCEL⟩ to cancel the command.

DELETE LINES

The DELETE LINES command lets you delete one or more lines from the form, including any fields or constant text the lines may contain. DELETE LINES works similarly to INSERT LINES. To use it, first move the cursor to the line you want to delete and then issue the command. The current line becomes highlighted, and you can use ⟨ARROW-DOWN⟩ or ⟨ARROW-UP⟩ to highlight more lines. When you press ⟨ENTER⟩, the highlighted lines are deleted; if you press ⟨CANCEL⟩, they aren't deleted.

Changing Whole Forms

The commands described in this section change the entire form. You should be cautious when you use them, since you will lose any work you have done designing the form.

DETAILED and COLUMN

These commands work much as they do in View mode and elsewhere in Cornerstone. Chapter 3, "Viewing Your Data," explains how they operate.

There are a couple of minor differences with these commands in Format mode. If you issue either COLUMN or DETAILED when the cursor lies within a subform, the command affects only the subform. Also, when you use the COLUMN command, Cornerstone doesn't automatically create headers the way it does in View mode, unless you issue COLUMN when the cursor lies within a subform.

DETAILED ALL-ATTRIBUTES is often a useful starting point when designing a form, since it gets you all the attributes in the file. You can then delete any fields you're not interested in.

DELETE FORM

If you decide that your form is a mess and you want to start over from scratch, use DELETE FORM. This command replaces whatever form you had with a blank one.

Manipulating Subforms

None of the following commands are available when you are designing subtotal, total, header, or title forms, since these forms can't have subforms in them.

INSERT SUBFORM

To add a subform to your form, move the cursor to the spot where you want to put the subform and issue INSERT SUBFORM. You must select a subrecord attribute from the menu, and you have the option of inserting the subform BEFORE or AFTER the cursor.

Cornerstone will insert a blank subform (one with no fields or text), so you have to use DETAILED, COLUMN, or INSERT FIELD to create some (sub)fields in the form. You can tell the cursor is in the subform when the word SUBFORM appears in the status line. The name of the subrecord is also displayed as part of the field name.

DELETE SUBFORM

To delete a subform, move the cursor anywhere inside the subform and use DELETE SUBFORM.

The CALCULATE Command

The CALCULATE command lets you add a new derived attribute or change an existing one. It works just as REPORT CALCULATE does in View mode, except that after you issue the command, you have a chance to move the cursor to the spot where you want to add the formula field. Format mode is a good place to use CALCULATE, since you can get the field to appear where you want it. (In View mode, Cornerstone just tacks the field to the end of the record form.) Chapter 3, "Viewing Your Data," explains how this command works.

Printing

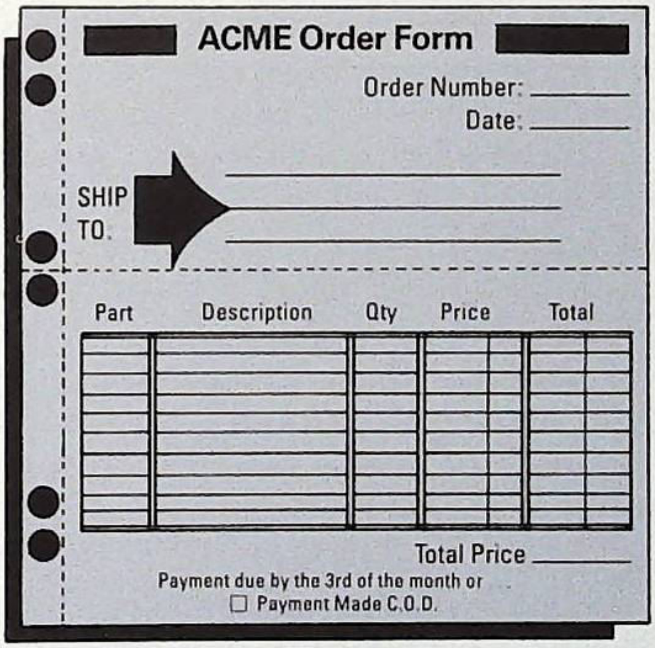
If you want to print a copy of your form, issue the PRINT FORM command. Cornerstone will print the form as it appears in Format mode, with underbars appearing in the fields instead of data.

In Format mode, you can also print a *grid,* which is a set of rules you can use as a design aid in creating Cornerstone forms for printing on pre-printed forms. To use this feature, insert the pre-printed form in your printer and issue the command PRINT GRID. Cornerstone will print the grid right over your pre-printed form. You can then use the row and column numbers in the grid to help you locate fields in the right place. The Format mode status line also shows row and column numbers.

Pre-Printed Forms

You can use Cornerstone to print information on pre-printed forms. This section describes how to do this.

Most pre-printed forms allow only a specific amount of space for each field. This is especially a problem for subrecords, which are generally multi-valued. Getting Cornerstone to print information on pre-printed forms is a two-step process. First, you design an appropriate Cornerstone form. Second, you answer "Yes" to the question *Use pre-printed form?* when you set the print characteristics in Print mode. This section shows you how to design the form.



ACME Order Form

Order Number: ________

Date: ________

SHIP TO:

Part	Description	Qty	Price	Total

Total Price ________

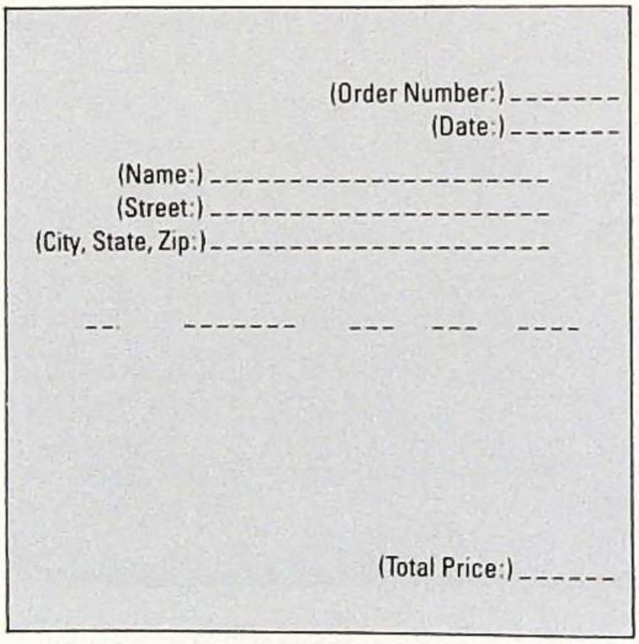
Payment due by the 3rd of the month or

□ Payment Made C.O.D.

Old-fashioned order form

Let's see how one Cornerstone user, Fred (from Omnifex Auto Parts), used Cornerstone to print orders on the pre-printed forms he got from his uncle Richard. The form Fred got (shown above) allows space for exactly ten line items. Since some orders will contain fewer than ten line items, and others might contain more, the problem is to make Cornerstone always use exactly ten lines for line items.

The figure below shows a Cornerstone form in Format mode that Fred designed to fit the Acme forms. The form has nine blank lines following the line items subform. These lines tell Cornerstone to reserve space for line items. The field names (in parentheses) are only included for reference. When Fred tells Cornerstone to print his orders using the pre-printed forms, the field names are not printed, since they already appear on the pre-printed form.



```
                          (Order Number:)_______
                                  (Date:)_______

          (Name:)_____________________
        (Street:)_____________________
(City, State, Zip:)_____________________


   --     -------     ---   ---    ----






                           (Total Price:)______
```

Cornerstone form designed to print on the Acme pre-printed order form (see previous page)

When Fred uses this form, and answers "Yes" to the *Use pre-printed form?* question in Print mode, Cornerstone prints the form in a special manner.

- □ Every field in the form starts on the row that it appears on in Format mode. For example, since the *Total Price* field appears on line 21 of the form, it will always be printed on line 21 on the pre-printed form, regardless of how many line items there are.
- □ Blank lines are interpreted as space holders for multi-valued fields and subrecords that precede them. If there are fewer values or subrecords than there are lines, Cornerstone will leave blank lines; if there are more, Cornerstone will not print the extra values or subrecords. In the latter case, Cornerstone will truncate the value(s) without warning. If you intend to use pre-printed forms frequently, you may wish to set the *Maximum number of values* characteristic for your subrecord attributes. In the example, Fred should set the maximum number of values for *Line Items* to 10. This way, Cornerstone will prevent him from creating an order with more than 10 line items, and he will never have to worry about an order being truncated because it doesn't fit in the form.
- □ Field names are not printed, regardless of the setting of *Display name?* in the field characteristics form. This is because pre-printed forms usually have the field names already printed on them. You can always include a field name as constant text if you want it printed on the form.

Subtotal, Total, Header, and Title Forms

There are some differences in the way Format mode works with a subtotal, total, header, or title form. For the sake of discussion, we classify subtotal, total, header, and title forms together as *summary forms*.

INSERT FIELD
This is the only command that operates differently for summary forms. A field in a summary form is used to display aggregate information, rather than an attribute. Therefore, instead of selecting an attribute when you insert the field, you select an aggregate function and an attribute, just as you do with the REPORT TOTAL command in View mode. The aggregate and attribute determine the value that appears in the field. You can learn more about aggregate functions in chapter 6, "Advanced Reporting."

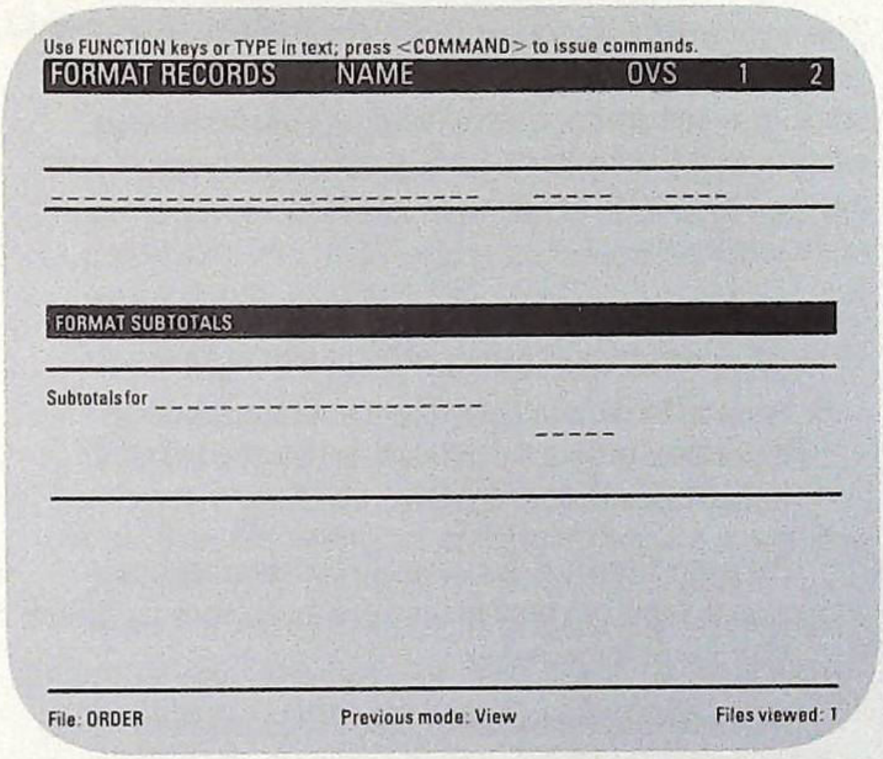
The following commands are *not* available in Format mode when you are working with summary forms.

- □ COLUMN
- □ DETAILED
- □ CALCULATE
- □ INSERT SUBFORM
- □ DELETE SUBFORM

Working with Two Forms

Sometimes when you design a report format, you'll want to make fields in different parts of the report line up. For example, you might want to make a field in the record form line up with a field in a subtotal or total. To make it easier to do this, Cornerstone lets you format two forms simultaneously, using the REPORT FORMAT command in View mode. For example, to format both the record and subtotal forms, issue the command

```
REPORT FORMAT RECORDS SUBTOTALS
```



```
Use FUNCTION keys or TYPE in text; press <COMMAND> to issue commands.
FORMAT RECORDS        NAME                        OVS      1      2

--------------------------    -----    ----

FORMAT SUBTOTALS

Subtotals for ---------------------
                                   -----

File: ORDER              Previous mode: View              Files viewed: 1
```

The figure above shows what Format mode looks like with two forms. Each half of the screen is a self-contained Format mode that works just the same as when you're only designing a single form. To switch from one form to the other, use the OTHER-FORM command.

Leaving Format Mode

When you're done designing your form, press the ⟨DONE⟩ key to leave Format mode. This will return you to whatever mode you were in before you entered Format mode. Any changes you made to the form are reflected when you return. Be sure to use the SAVE command in View, Update, or Select mode to save your changes, otherwise the result of your efforts may be lost.

Tips for Form Designers

This section provides some suggestions that we hope will give you ideas for designing your own forms. They cover some common situations that arise. As you become an experienced Cornerstone user, you'll probably invent your own techniques for designing forms to meet your particular reporting needs.

Constant Text

Here are some ways of using constant text in forms:

As a title. You can use text to create a name or letterhead in your form, such as "Tweedle-dee Consultants, Inc.—Client profile."

As instructions. You can use constant text as instructions or commentary in a form, such as "Enter your name here -------> ".

As text in a form letter. You can intersperse fields and constant text to get a form letter effect. For example, you might begin a form with "Dear ________,". (The underbars represent the *Customer Name* field, say.) Another trick you can use in title or header forms is to create a line such as
"Orders from ______________ to ______________"
where the two fields might be defined as MIN*(Order_Date)* and MAX*(Order_Date)*. With this technique you generally don't want the field names displayed.

To draw lines. You can use rows of dashes, equal signs, asterisks, or any other character to draw lines that can be used as separators in the form. This technique is used in the *Fine* format for the *Order* file that comes with the Sample Database.

As bullets for subrecords. If you type a character, such as '*' or '>', at the left of a subform, it will appear next to every subrecord.

Common Mistakes

Typing underbars (_) to lengthen a field. If you type the character underbar (_) into a form, this character will be displayed as is. Typing underbars does not create or lengthen a field. To create a field, use INSERT FIELD. To lengthen a field, use ADJUST-WIDTH or SET-CHARAC.

Inserting a line in a form versus a subform. When you use the INSERT LINES command with a form that contains subforms, the lines are inserted in the subform if the status line indicates SUBFORM when you issue the command; otherwise they are inserted in the (parent) form—with the following exception: When the last line in a subform is also the last line in the parent form, inserting lines AFTER the last line in the subform causes the lines to be inserted in the *parent form.* To insert lines at the end of the *subform,* first insert a line at the end of the form, then move inside the subform and INSERT LINES AFTER the last line. You can then delete the last line in the parent form.

When to Use Format Mode

Below are a few situations when Format mode is useful.

When COLUMN and DETAILED aren't what you want. You may be dissatisfied with the position of the columns or headers in a column format. If so, you can issue the command REPORT FORMAT HEADER RECORDS to format the header and record forms. You can move, and change the width of, the fields in the records form, and you can adjust the header so that the field names remain aligned with the fields. You can also use Format mode to add a column to a column form.

To design special formats. You can use Format mode to create customized formats. You can type your business logo in the form, or draw lines to separate subtotals and totals.

To design a form to use with a pre-printed form. You can use Format mode to design a (Cornerstone) form for printing information contained in your database on a pre-printed form.

To customize a data entry (update) form. If you (or others who use your system) like to enter data in a particular order or format, you can design a data entry form that mimics this format. See chapter 4, "Entering and Updating Records."

The COLUMN and DETAILED commands in View mode suffice for most reporting needs. You can use Cornerstone without ever entering Format mode, but once you do, you'll probably want to customize all your formats. Feel free to experiment with different designs. Format mode will bring out the artist in you.

PLANNING YOUR DATABASE

9

CHAPTER

CHAPTER 9

CONTENTS

Planning Your Database

Introduction

In this chapter, you'll learn how to design your own database. With Cornerstone, you're the architect. You decide what information is important. You decide how to organize it. And, if you're thinking, "Me? Design a *database*?"—read on. Soon you'll have plenty of ideas about how Cornerstone can help you manage your information.

We'll show you how Fred designed his Cornerstone database to keep track of customers and orders for his auto parts business. You'll see how he used various features of Cornerstone to streamline his operation and allow him more time to spend fixing up his '71 Plymouth Duster.

Using Fred's system as an example, we'll teach you about files, attributes, relationships, and a few other database concepts you'll need to know to design your database successfully and get it running.

Our chapter begins by exploring just what a Cornerstone database *is*.

What's a Database?

A database is a collection of *files*. Each file contains *records*. The structure of a record is defined by its *attributes*, each of which has a *value*.

Let's explore these essential elements one at a time.

□ A database is a collection of files.

Consider what goes on at Fred's auto parts company, Omnifex. Fred used to run his business from a three-drawer steel filing cabinet. One drawer was labeled "Customers," another "Orders," and the third, "Parts."

When Fred got Cornerstone, he made these the basic files in his database: a *Customer* file, an *Order* file, and a *Part* file.

□ Each file contains records.

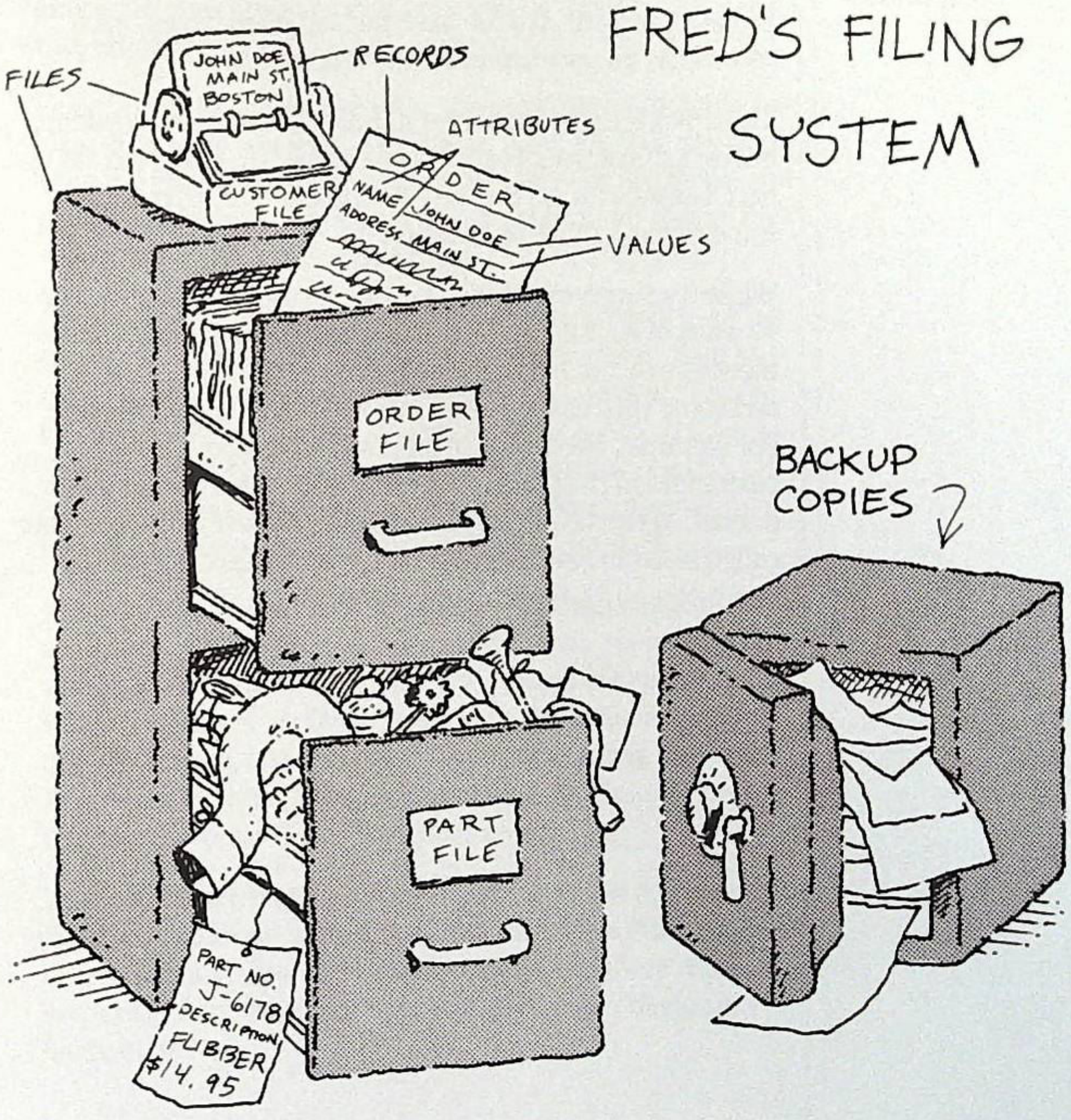
Fred has a fair number of reliable customers who order parts. Whenever a customer calls to place an order, Fred pulls out an order form and takes down the necessary information. In his old order entry system, he would then put a copy of the order in the file cabinet. Each order was a separate sheet in Fred's filing cabinet. Now, each corresponds to a separate *record* in his *Order* file. The same is true for customers and parts. At last count Fred stocked 74 different parts, so his *Part* file contains 74 records, one for each part. Likewise, each record in Fred's *Customer* file contains information about a particular customer.

□ The structure of a record is defined by its attributes.

Every record in a file has room for the same pieces of information, just like a pre-printed paper form. These pieces of information, called *attributes,* constitute the record. For example, the records in Fred's *Customer* file have attributes *Name, Street,* and *City,* to name a few. These and the other customer attributes together define a customer for Fred's purposes.

Fred's *Part* and *Order* files have their own attributes, which are different from those he defined for his *Customer* file.

Attributes are the building blocks that determine what information is contained in the records in a file. They describe entities such as customers or parts, and typically correspond to properties of real-world things (which is why they're called attributes). You can create whatever attributes you want to describe an entity. You decide which attributes are important and which aren't. Fred could have included sex, birthday, and marital status in his customer file, but he decided these weren't important for his business needs.



FRED'S FILING SYSTEM
FILES
JOHN DOE
MAIN ST.
BOSTON
RECORDS
ATTRIBUTES
CUSTOMER FILE
ORDER
NAME JOHN DOE
ADDRESS MAIN ST.
VALUES
ORDER FILE
BACKUP COPIES
PART FILE
PART NO. J-6178
DESCRIPTION FLUBBER
$14.95

□ Each attribute has a value.

While all records in a file have the same attributes, each record has its own *values* for the attributes.

Thus, while every customer in Fred's database has *a* name, not every customer has the *same* name. (In fact, Fred set up his *Customer* file so that no two customers *can* have the same name—more on this later.)

While it's simpler to think of every record as having a value for each of its attributes, this is not necessarily true. Some records may not have any value for a particular attribute, and some (discussed below) can have more than one value. For example, Fred used to jot down notes about his customers in the margins of the sheets in his Customers drawer. When he set up his database, he defined an attribute called *Notes* in his *Customer* file, in which he can enter as many notes as he wants, or none at all.

Files and Attributes

The key step in defining your database is to identify the basic entities in your application. These will be the *files*. For Fred, they were customers, orders, and parts.

Once you've identified the files, the next step is to figure out what attributes you want each file to have. How do you know what attributes to include in your design? The attributes you select will depend on your particular application, that is, on what you want your database to *do*.

For example, Fred knew he wanted to create reports on his customers grouped by city and zip code, so rather than just creating an attribute called *Address* in his *Customer* file, he broke the address up into several attributes: *Street, City, State,* and *Zip*. Because he did this, he can now select customers from, say, Cambridge, or produce reports containing customer information grouped by state.

You may already have a paper system that can suggest attributes. Before Fred got Cornerstone, he took orders using the Acme Order Forms he got from his uncle Richard. When he defined his *Order* file, he used the fields in the order form (such as *Customer Name, Line Items,* and *Total Price*) as attributes. (You may have seen database systems that use the word "field" to denote an attribute.)

You don't have to figure out all the attributes in advance, since Cornerstone will let you add more, even after you've entered data in your files. Nevertheless, the more attributes you can identify early on, the more useful your design is likely to be.

Every Attribute Has a Type

In Cornerstone, every attribute has a *type*. The type of an attribute indicates what type of information is stored in the record for the attribute. Below is a summary of Cornerstone types.

String. The data is textual, such as a name, a city, or notes. The term comes from the phrase "a string of characters."

Number. The data is a number that may have a decimal part (for example 3.78). A typical example of a number attribute is a dollar amount, such as *List Price* or *Price After Discount.*

Integer. The data is an integer; that is, a number with no decimal part, such as 1, 54217, or -17. (Cornerstone makes no internal distinction between numbers and integers. The difference lies in how the value is displayed and in what values may be entered.)

Date. The data is a date, such as 10/9/1959 or Friday, October 9, 1959.

Time. The data is a time, such as 5:17 PM or 17:17.

Enumerated. The data is one of several values, which you specify when you define the attribute. The values the attribute can have are called the *enumerated values*. For example, the payment terms for an order can be Net30, Net60, COD, or Cash.

Boolean. The data is either Yes or No. For example, you could create a *Questionnaire* file that had boolean attributes in answer to questions such as "Ever been arrested?" and "Ever suffered a heart attack?"

Derived. The value of the attribute is found from other information in your database. Derived attributes are discussed in detail in a later section.

Subrecord. The attribute represents a repeating group of information. Subrecords are discussed in detail in a later section.

The type determines *how you can use* a particular attribute—for example, what calculations you can make, and how the attribute is sorted. You can total a column of numbers or integers, but not a column of dates. Dates are sorted in chronological order, as are times, but strings are sorted in alphabetic order.

The type also determines what values are allowed for a particular attribute. For example, Cornerstone won't let you enter "10 Claflin Road" for a number attribute, nor "12/32/84" for a date.

Besides restricting what information you can put *into* your database, the type also determines how it comes *out*. For example, dates are displayed using slashes, and times are displayed with a colon between hours and minutes. (You can select other display styles, if you wish.)

The type also tells Cornerstone how to store the data in the file. This is not something you need be concerned with, but it helps keep your database trim. For example, Cornerstone doesn't store the actual value for an enumerated attribute, only a numeric code. This makes no difference to you—since Cornerstone translates the code into one of the enumerated values whenever the value is displayed or printed—but it saves disk space.

What Type Should I Use?

When deciding what type an attribute should be, the most important question to consider is, "How do I plan to use the attribute?" Do you plan to *sort* using it? Do you want to make *calculations* based on the attribute? What does the attribute *represent* in the real world?

This last question is often a good start, but it doesn't always lead to the best answer. For example, you might wonder why Fred made *Zip* a string instead of a number or integer—after all, isn't a zip code just a number? If Fred had made *Zip* a number, then "02139" would display (and print) as "2139"—and what about the new 9-digit zip codes that contain a hyphen? Besides, Fred knew he wouldn't have to do arithmetic with his customers' zip codes!

Strings are the simplest kind of raw data, but, if you plan to perform arithmetic calculations using an attribute, you should make it either a number or an integer. With numbers and integers you can use the various display styles Cornerstone provides. For example, you can make numbers display with dollar signs and with commas separating thousands, and you can make negative numbers appear in parentheses.

It's obvious when to use dates and times. To make things easy, Cornerstone knows the expressions TODAY and NOW. TODAY is used to automatically fill in the current date when you create a new record; NOW gives the current time. You can even type expressions like LAST WEEK when you're filling in a date value. The chapter "Entering Information Into Forms" in *Owner's Handbook II* lists all the expressions you can enter for dates and times.

Enumerated attributes are useful when you want an attribute's value always to be one of a set of values—for example, payment terms that can be Net30, Net60, COD, or Cash. The enumerated values can be whatever you want, but if you choose them so they can be distinguished by one or a couple of characters, it will be easier to enter them when you're filling out a form.

A trick you can use is to letter the enumerated values. For example, instead of having enumerated colors "Red," "Rouge," and "Rose," you could make them "A) Red," "B) Rouge," and "C) Rose." Then all you have to do is type a single letter when you're entering the color in a form.

A boolean attribute is a special case of an enumerated attribute that has two values: Yes and No. You can use boolean whenever you know these are the only values an attribute can have (such as in the questionnaire example above).

Derived and Initial Values

Often one of the attributes in a file can be computed from other information using a simple rule. For example, in Fred's Order form, the price for each line item is just the price of the part multiplied by the quantity ordered.

There are two ways you can get Cornerstone to compute the value of an attribute. One is to make the attribute derived. The other is to give it an initial value expression.

Derived Attributes

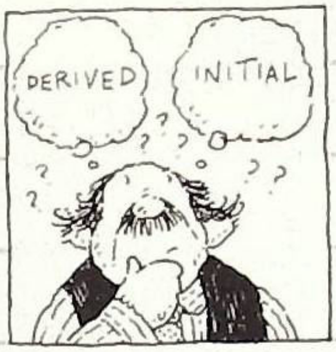
The form below shows how Fred defined the *Price* attribute. He made *Price* be of type Derived, which tells Cornerstone that its value is derived from other information in the database.

```
Name: PRICE
Description: The price is derived by multiplying the quantity by the
             list price.
Type: DERIVED
Derivation: QTY * LIST_PRICE
```

Whenever you define a derived attribute, you must tell Cornerstone *how* the value is to be derived. You do this by supplying a *derivation expression.* In the example, the derivation expression "Qty * List_Price" tells Cornerstone to compute *Price* by multiplying *Qty* (quantity) and *List Price.*

Before discussing derivation expressions further, let's pause to see what Fred has gained by making *Price* derived. Now, Fred never has to figure out the price of a line item. If a customer calls back five minutes after placing an order to say he wants 20 more vacuum modulator wrenches, all Fred has to do is change the quantity—Cornerstone does the rest. This saves Fred time, reduces the likelihood of errors, and makes it possible to give the customer the new price immediately. Cornerstone also figures out the other derived attributes Fred defined, such as *Total After Discount.*

There are many kinds of derivation expressions, and many special functions you can use in them. Some derivation expressions refer to information in another file, some refer to information in a subrecord or parent file. The chapter "Entering Information into Forms" in *Owner's Handbook II* lists the valid kinds of derivation expressions.



Derived Versus Primitive

Derived is not a *type* in the same way that, for example, String and Number are types. In the example, *Price* is derived—but it's also a number. Attributes that are not derived (such as strings and numbers) are called *primitive* to distinguish them from derived attributes. All derived attributes are ultimately computed from primitive ones.

The main difference between derived and primitive attributes is that no data is stored for derived attributes. The value for a derived attribute is computed "on the fly" every time it is displayed or printed.

Initial Values

Sometimes you want the system to compute a value only when you first create a record, either because you don't want the value to change once it's created, or because you want the option of changing it yourself.

For example, Fred got tired of filling out the current date every time he took an order, so he made the *Order Date* attribute have an initial value of TODAY (this is one of the functions you can read about in "Entering Information into Forms" in *Owner's Handbook II*). Now whenever Fred takes an order, Cornerstone automatically fills in the current date. Fred can still change the date if he wants—for example, if he's entering the previous day's orders.

Derived or Initial Value?

How do you decide whether to make an attribute derived or to use an initial value?

Let's look at the *List Price* attribute in Fred's *Order* file. Fred set up *List Price* in his *Line Items* to have an initial value that's computed from the current list price in his *Part* file.

(Fred was able to do this because he defined a relationship—more about relationships in a moment.) All Fred has to do when he takes the order is enter the part number. Cornerstone fills in the price for him.

If Fred had made *List Price* derived, whenever he raised the price of one of his parts, the *List Price* for that part in all his orders would also change—even for orders that have already been paid and delivered! Because Fred used an initial value, the list prices in his orders never change once he's entered them (unless Fred explicitly changes them himself). This allows him to keep a permanent record of all his orders.

The following table summarizes some of the differences between derived values and initial values.

DERIVED VALUE	INITIAL VALUE
Value is not actually stored in the record. Requires no disk space.	Value is stored in the record. Requires disk space.
Value is computed every time it's needed. Value is always up-to-date.	Cornerstone never recomputes the value once it's created, unless you tell it to. *
You can't change the value in Update mode.	You can change the value in Update mode.
Useful if value is continually changing.	Useful if you don't want the value to change, unless you explicitly re-enter it.

* You can use the RECOMPUTE utility to recompute values for primitive attributes having an initial value expression. You can learn more about this feature in *Owner's Handbook II,* chapters 4 and 5.

If you can't decide whether an attribute should be derived or have an initial value, don't worry: with Cornerstone, you can always switch between derived and initial values, even after you've entered data into your files.

Getting it Together: Relationships

Okay, so you've figured out what files you want, and you even have a pretty good idea what attributes you want them to have. So far you've got a bunch of unrelated files. Now you must tell Cornerstone how you want them to fit together.

☜

You do this by defining *relationships* between files in your database. ☞ A relationship is a link between two files that allows information to be shared between them. ☜ Once you define a relationship between two files, you can define derived attributes and initial value expressions in one file that refer to information in the related file. Relationships let you harness many of the powerful features of Cornerstone and turn your database into a unified, smooth-working system.

Fred's order entry system shows how useful relationships can be. Whenever Fred takes an order, he types in the customer name, and Cornerstone automatically finds the customer's address and puts it in the order form. How did Fred set up his database to do this?

Both the *Customer* file and the *Order* file contain an attribute called *Name* that identifies a customer. Fred defined a *relationship* between the *Name* attribute in the *Customer* file and the *Name* attribute in the *Order* file. (The attributes need not have the same name; Fred chose to use the same name just to remind himself that the attributes are related.)

Defining the relationship between the *Customer* and *Order* files allows Fred to share information between the files.

Using Relationships to Share Information

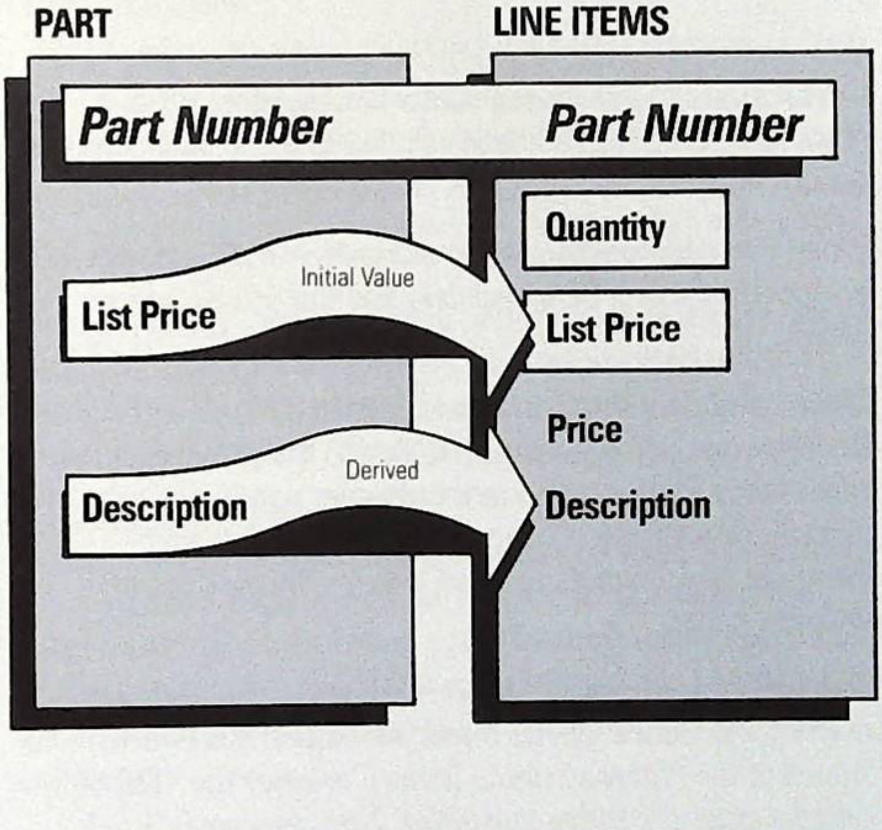
Once Fred defined a relationship between his *Customer* and *Order* files, he defined the attributes *Street, City, State,* and *Zip* for his orders to be derived from their counterparts in the *Customer* file. For example, he defined *Street* to be derived, with derivation expression "Customer.Street". "Customer" stands for the *Customer* file; "Street" stands for the *Street* attribute in the *Customer* file; the dot (.) separates the two names. Fred defined *City, State,* and *Zip* similarly.

The expression Customer.Street tells Cornerstone how to compute the value for *Street* in the *Order* file. It says, "find the related customer record in the customer file and get the street address from it." Cornerstone knows to look for the customer with the same *name* because Fred defined the relationship to be between *Customer.Name* and *Order.Name.*

Now that Fred has things set up the way he wants, he never has to type the address in an order form, since Cornerstone fills it out for him. He just enters it once in the *Customer* file whenever he gets a new customer.

If you followed our earlier discussion about derived attributes, you may be thinking, "That's fine—but what if one of Fred's customers *moves?* Won't all the old orders be changed?" Well, you're right. Fred thought about this and considered making *Street, City, State,* and *Zip* primitive attributes with initial values (instead of being derived), but in the end he decided he didn't care to keep a permanent record of his customers' old addresses in his orders. The space he saved was more important to him. Besides, the last time one of his customers moved was in 1953, and that was to Alaska.

Fred defined another relationship, between his *Line Items* subfile and his *Part* file, based on the *Part Number*. Then he set up the *Description* attribute in *Line Items* to be derived as "Part.Description". Now, when Fred enters a part number for an order, Cornerstone finds the description of the part and displays it in the order form. This way Fred knows he hasn't typed the wrong part number.



Keeping Your Database Consistent

What would happen if Fred forgot to enter a name for some customer? What would happen if he accidentally entered two customers in his *Customer* file that had the same name? Which address would Cornerstone use when he enters the name in an order form?

And suppose, while taking an order, Fred tried to enter a name that isn't the name of one of the customers in his *Customer* file. How would Cornerstone fill out the address now?

You can find the answers to these questions by using Cornerstone to see for yourself what happens. Fred, however, set up his database so none of these situations can occur.

When Fred defined the *Name* attribute in his *Customer* file, he specified that it be *mandatory* and that it have *unique values*.

If Fred ever forgets to enter a customer's name, Cornerstone will remind him to do so. And, Cornerstone won't allow Fred to add a new customer name that's already in his *Customer* file. Cornerstone ensures that every customer has *a* name, and that every customer has a *different* name.

To handle the second situation, Fred told Cornerstone to restrict the values for the *Name* attribute in his *Order* file to values of the *Name* attribute in his *Customer* file. This is called a *cross-file range constraint.* Now, whenever Fred enters the *Name* in an order form, Cornerstone checks that the name he enters is the name of one of the customers in his *Customer* file.

The range constraint does more than just restrict Fred's input. It lets him see his customers' names by pressing the ⟨OPTIONS⟩ key when he's filling out the *Name* in an order form. This relieves Fred from having to remember the names of all his customers. He doesn't even have to type the whole name—for example, the only customer Fred has whose name begins with "W" is Watertown Car Center, so to take an order from them, Fred just types "W⟨ENTER⟩." Cornerstone fills in the full name. All this happens because Fred restricted the *Name* in his *Order* file using a cross-file range constraint.

☒ Cross-file range constraints are similar to enumerated attributes. Just like an enumerated attribute, an attribute with a cross-file range constraint can only have one of a set of values—but the values it can have are stored in a separate *file* instead of being defined at the time you define the attribute. The advantage of using a file is that you can associate other information with the value (such as the address and other information associated with a particular customer). Also, you can have many more allowed values than with an enumerated attribute.

The mandatory, unique values, and range constraints can be used with *any* attributes—not just ones involved in a relationship. Nevertheless, these constraints were designed with Fred's particular situation in mind.

Evaluating Relationships

To summarize, relationships allow files to share information. They go hand-in-hand with derived attributes and initial value expressions. Defining a relationship merely specifies a *link* between two files. You *use* the relationship to extract information by defining derived attributes and initial value expressions.

As we saw with Fred's database, you can use the mandatory, unique values, and cross-file range constraints to ensure that information in different files is consistent.

Multi-valued Attributes

With Cornerstone, attributes can have more than one value. For example, Fred likes to keep a list of people to contact at his customers' businesses, so he created an attribute called *Contacts* in his customer file. Because Fred often has several contacts with the same customer, he made *Contacts multi-valued.* Cornerstone allows Fred to have several contacts at each customer site.

There are other uses for multi-valued attributes. Remember Fred's *Notes* attribute? It's multi-valued too. Because he made *Notes* multi-valued, Fred can type several notes about a customer.

Listed below are a few facts about multi-valued attributes.

- □ You don't have to worry about space requirements for multi-valued attributes, since Cornerstone uses only as much space as is required for the values you enter. If you don't enter any values, no space is used.
- □ You can use the aggregate functions (such as SUM, MINIMUM, and AVERAGE) with multi-valued attributes. These functions (listed in "Entering Information into Forms" in *Owner's Handbook II*) allow you to derive information computed from all the values taken as a whole.
- □ You can use the FIRST and LAST functions to copy the first and last values to another file.
- □ You can sort a report, or group subtotals, using a multi-valued attribute, but only the first value is used.

While you can do things with multi-valued attributes that you couldn't do with single-valued attributes, Cornerstone imposes some limitations in how you can use them.

You can't use a multi-valued attribute to define a relationship (how would Cornerstone know which value to use?), and you can't share multiple values across files using a derivation expression, the way Fred copied the customer address from the *Customer* file to the *Order* file.

Subrecords and Subfiles

One of Cornerstone's distinguishing features is that it lets you define files that have *subrecord* attributes. What's a subrecord? We can find the answer to this question (not surprisingly) by looking at Fred's database.

Fred's order forms contain a line for each item ordered. Each line item has a part number, a description of the part, the quantity ordered, the list price, and the total price of the line item.

The first thing Fred noticed about the line items is that they form a repeating group of information. Each line item is made up of the same pieces of information.

Fred figured that line items should somehow be an attribute in his *Order* file. Because each order can have several line items, he supposed the attribute should be multi-valued—but he couldn't figure out what *type* line items should be. String? Well, the part number and the description are strings, but then price is a number. It seems that none of the types we've discussed so far is quite right. As you've probably guessed by now, *subrecords* provide the solution to Fred's problem.

Fred defined the *Line Items* attribute to be of type *subrecord.* Cornerstone then automatically created a *subfile,* which it named *Line Items,* after the subrecord attribute. Fred then defined attributes *Part Number, Description,* etc., in the subfile. When Fred enters the lines items for an order, they go into the *Line Items* subfile. The records in this file are called *subrecords*. Although all the line items Fred enters for any order go into the same file, *Line Items*, each subrecord in the *Line Items* file belongs to a particular order.

Because the line items belong to the *Order* file, the *Order* file is called the *parent file,* and the order containing a particular group of line items is called the *parent record*. The attributes in a subfile are called *subattributes* when it's necessary to distinguish them from attributes in the parent file.

☒ For those who love terminology, we provide the following more concise definitions. An attribute of type *subrecord* is an attribute whose values are records. These records are called *subrecords*. (The term *subrecord* thus denotes both an attribute *type* and a *kind of record*.) The subrecords are stored in a file whose name is the same as that of the attribute. This file is called the *subfile*. The attributes of the subfile, sometimes called *subattributes*, determine the structure of the subrecords.

Other Examples of Subrecords

Line items is a prime example of a subrecord because it illustrates many of the properties that are characteristic of subrecords; however, two more examples may serve to further clarify when to use subrecords.

***Employment History* in a *Personnel* file.** Each employee in a company may have had several previous jobs. Each job description has the same structure: dates of employment, company name, location, salary, and so on. A database could be designed for this situation by defining an *Employee* file with a subrecord attribute called *Employment History*. Dates of employment and the rest would be attributes of the *Employment History* subfile.

Contacts in a Client Tracking file. You might create a *Client Tracking* file to keep track of meetings with your clients. One way do do this is to define a subrecord attribute, say, *Contacts,* with attributes such as *Date of Contact, Person Contacted, Telephone Extension,* and *Topics Discussed* (this last could be multi-valued, like Fred's *Notes*).

The common thread in these examples is that the subrecords form a repeating group of information.

Properties of Subrecords

In each of the examples, the subrecords *belong* to the parent record. *Line Items* belong to a particular *order; Employment History* belongs to a particular *employee; Contacts* are with a particular *client.*

The sense in which subrecords belong to their parent record is more than metaphorical: ☞ subrecords can only be created or changed along with the parent record, and when you delete the parent record, all of its subrecords are also deleted. ☜

For example, Fred can't create a line item by itself, but only in some order. If Fred deletes the order, all of its line items are also deleted.

A file can have more than one subrecord attribute. You can have up to eight subrecord attributes in a single file.

A subfile can't have subrecords; you can't have a subrecord within a subrecord.

The Relationship Between a File and its Subfiles

Whenever you create a subrecord attribute in a file, you can share information between the parent file and the subfile in the same way you could between two related files.

For example, Fred defined *Total Before Discount* in his *Order* file to be derived, with derivation "SUM(*Line_Items.Price)*".

This expression tells Cornerstone to compute *Total Before Discount* by adding up the prices for all the line items in the order. (It's an example of a subrecord aggregate expression.)

When to Use a Subrecord?

Any time you have a bunch of attributes (often called a *repeating group*) that go together and are likely to appear more than once—such as the attributes that describe a line item, or those that describe an employee's previous job—you should consider combining the attributes into a single subrecord attribute.

Whether or not a subrecord is the right choice depends on what you want to do. As you make your decision, keep in mind the differences and restrictions discussed above.

You can create a single-valued subrecord if you wish, but this is generally not useful. You can achieve the same effect by simply making all the subattributes be attributes in the parent file.

For example, you might consider making an address a subrecord attribute, with subattributes *Street, City, State,* and *Zip*. In fact, this is what Fred did the first time he set up his database. He soon discovered, however, that he could not sort his reports by *Zip,* nor could he compute subtotals by *State* and *City,* since these were subattributes. Fred realized his mistake and changed his database so that *Street, City, State,* and *Zip* are all attributes of the *Customer* file.

Fred's original idea might have been useful if the proposed *Address* attribute were multi-valued—for example, if he were keeping track of several addresses for each customer—but he would still have had to sacrifice the ability to sort on the subattributes.

A True Story

We conclude this chapter with a short example relating how one Cornerstone user combined several of the features we've been discussing. We hope it illustrates some of the possibilities for database design and provides you with inspiration for designing your own database.

R.W., a product manager at Infocom, uses Cornerstone to keep track of clients. Her database contains a file called *People*. R.W. likes to keep track of clients' first and last names, but she also must ensure that the people in her *People* file have unique full names—for example, that there are not two Elmer Fudds. R.W. first thought she would define a single attribute, *Full Name,* but she couldn't figure out how to then automatically extract a person's first and last names from his or her full name.

R.W.'s solution was as follows. She defined attributes *First Name* and *Last Name* in her *People* file. She then used the STRING function to combine these in a single attribute, *Full Name,* whose initial value is STRING*(First_Name,*" ", *Last_Name)*, and which she specified to be *mandatory* and to have *unique values*.

Now, when R.W. enters a new person in her *People* file, she enters the person's first and last names separately, and Cornerstone automatically fills in the full name. If the combined full name already appears in the *People* file, Cornerstone won't fill in the value, but will instead leave *Full Name* blank (since the initial value would violate the uniqueness constraint)—however, because R.W. made *Full Name* be *mandatory,* Cornerstone won't let her leave Edit mode without either entering an acceptable value for *Full Name,* or changing either *First Name* or *Last Name* so as to make the combined full name different from all other full names in the *People* file.

R.W. even made *Full Name* indexed, and used it to define a relationship with another file, *Companies*.

Although the situation in this example may seem unusual, it's actually not all that uncommon. The STRING function is often useful when you want the combined values of one or more attributes to be unique. STRING even allows you to combine attributes of different types, such as numbers, dates, and times.

Bon Voyage!

Database design can be the most difficult aspect of Cornerstone, since it requires you to form an abstract idea of the information you want to manage. We hope this chapter has provided enough guidelines to start you on the road to becoming a successful database designer.

Though you may spend a lot of time planning your database, you'll find that as you start *using* it, you'll be less concerned about the database definition and more about practical matters like creating a clear report in two minutes. The more care you take in designing your database, the easier such things will be.

The best advice we can give is this: experiment with your database using a small amount of data. Try to create the reports and selection criteria you think you'll need. If everything works the way you expect, *then* go ahead and enter 5 billion records; if not, you'll probably see a way to change your database that solves the problem.

Designing a database isn't difficult if you proceed thoughtfully, and you'll see in the next chapter that it's even easier to *build* your database once you've *designed* it.

DEFINING YOUR DATABASE

10 CHAPTER

CHAPTER 10

CONTENTS

Defining Your Database

Introduction

In this chapter, you'll learn how to use Define mode to create the database you've designed.

The previous chapter, "Planning Your Database," is a companion to this one. It explains some of the terms used here (such as files, attributes, and relationships), and also supplies guidelines for designing a database. Even if you're familiar with database concepts, we suggest you read "Planning Your Database" before building your database.

Once you have a clear idea of what files, attributes, and relationships you want your database to contain, you are ready to *define* your database to Cornerstone. Define mode is the mode where you do precisely this.

To set up a new database, use the NEWDB command (at the DOS prompt). On a two-diskette system, the Sample Database Copy diskette must be in drive A. See "Read This First" for complete information.

Overview of Define Mode

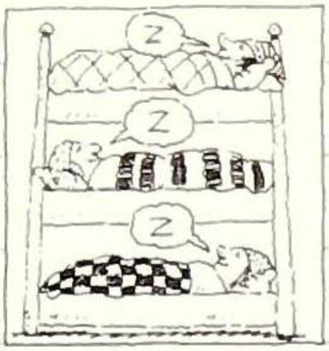
In Define mode, you tell Cornerstone how you want to structure your database. It's the place where you create and change the *definition* of your database. You can define and redefine files and attributes, and you can define and remove relationships between files.

You get to Define mode from Initial mode via the DEFINE command.

Three for the Price of One

Define mode has three levels (you can think of them as submodes if you like).

> **Top level.** This is where you start when you enter Define mode. The display shows a list of all the files in your database, and the menu contains commands that work with these files.



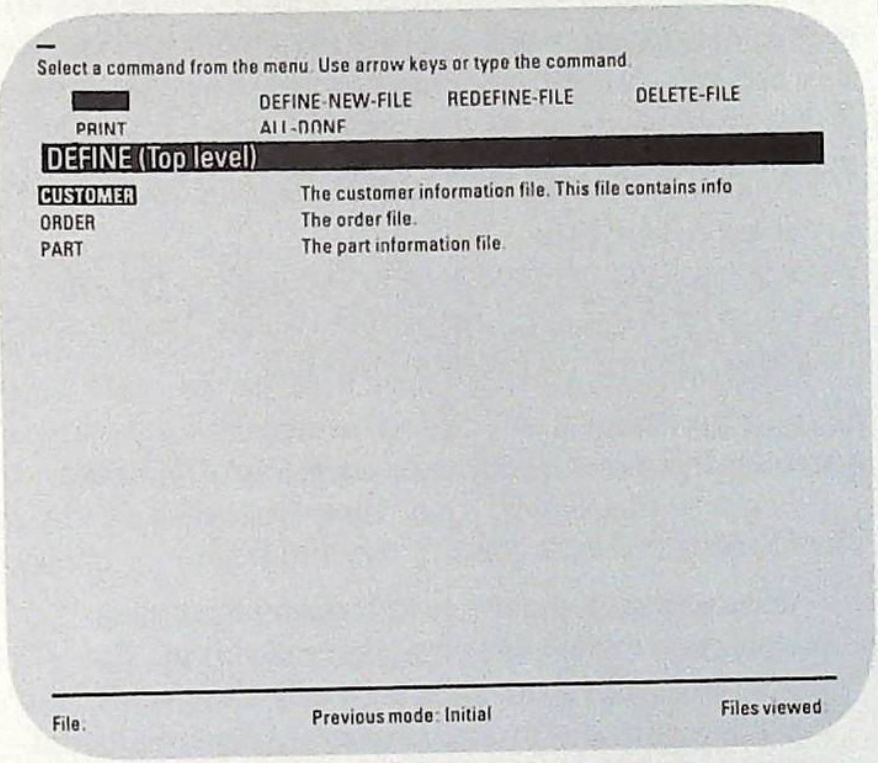
File level. You get to File level when you either define a new file or redefine an existing one. The file you define (or redefine) becomes the current file. The display shows a list of all the attributes in the current file, and the menu contains commands that work with these attributes.

Subfile level. You get to Subfile level when you define or redefine a subrecord attribute at File level. When you finish editing the attribute characteristics for a subrecord attribute, and press ⟨DONE⟩, Cornerstone takes you to Subfile level. The display lists all the subattributes in the subfile, and the menu contains commands that work with these subattributes. Subfile level is virtually identical to File level. The only difference is that the names of the commands contain the word "subattribute" instead of "attribute." This difference is intended only to serve as a reminder that you are at Subfile level, as there is no difference between attributes and subattributes.

LEVEL	COMMANDS WORK WITH
Top	Files in the *database*
File	Attributes in the current *file*
Subfile	(Sub)attributes in the current *subfile*

Each level contains similar commands—the difference is that the commands work with different objects. Cornerstone takes you naturally from one level to the next as you define or redefine files and subfiles, so you needn't be overly concerned with what level you're at. If you ever get lost, you can glance at the top status line or press the ⟨HELP⟩ key to see where you are.

The Display



```
_
Select a command from the menu. Use arrow keys or type the command.
                        DEFINE-NEW-FILE   REDEFINE-FILE        DELETE-FILE
   PRINT                ALL-DONE
DEFINE (Top level)
CUSTOMER                    The customer information file. This file contains info
ORDER                       The order file.
PART                        The part information file.




____________________________________________________________________________
File:                       Previous mode: Initial                 Files viewed
```

In Define mode, the display contains a list of the files or attributes (depending on what level you're at) available for you to work with. One of these is highlighted, and is the *current* file or attribute. Some commands use the current file or attribute as a default choice.

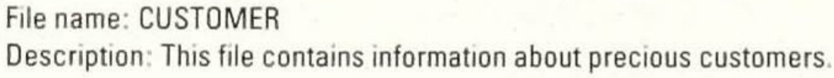
You can use the motion keys (such as ⟨NEXT⟩ and ⟨PREVIOUS⟩) to change the highlighted file or attribute, and you can use the scrolling keys when there are more items than fit in the display.

Files

When you enter Define mode, the menu contains commands that work with files. One of these commands, DEFINE-NEW-FILE, lets you define a new database file. Another, REDEFINE-FILE, lets you change the definition of an existing file. Both of these commands are used to get to File level, with either the new file or the existing one as the current file.

Creating a New File

When you select DEFINE-NEW-FILE, Cornerstone presents you with a form to fill out that defines the file. The following illustration shows what the form looks like.



File name: CUSTOMER
Description: This file contains information about precious customers.

The file definition form contains only two fields.

File name. You can generally enter any name you want, though there are a few restrictions. For example, you cannot enter a name you've already used as the name of a file, and you can't use certain special words such as NOW and TODAY. Cornerstone will inform you when you try to enter an illegal name.

Description. The description is a place to supply any text you want that describes the file, such as "This file contains recent employment history for employees working here." You aren't required to enter a description, but it's a good idea to do so, since Cornerstone uses the description in help messages relating to the file. This way, you can use file descriptions to create your own help messages.

When you've filled out the form, just press ⟨DONE⟩, and the file is defined! Of course, this is somewhat misleading, because although you've defined a file, you can't put any data in it until you define attributes for it.

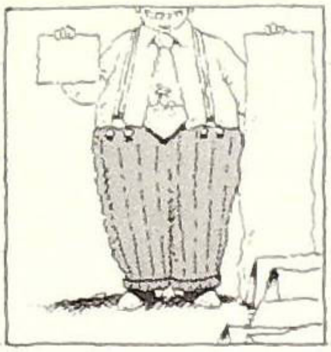
☒ If, for some reason, you decide while editing the form that you really don't want to create a file after all, use the CANCEL FORM command in Edit mode to return to the Top level menu without creating a new file.

Changing a File

Once you've defined a file, you may decide that you want to change it. You might want to change the name or description, or you might want to change the definition of one or more attributes in the file. The REDEFINE-FILE command lets you change a file. To use this command, first select the file you want to change. You then have two options:

> **FIRST-FILE-INFO-THEN-ATTRIBUTES.** If you select this option, Cornerstone will give you a chance to edit the form you filled out when you created the file. You can change either the file's name, its description, or both. After editing the form, the system enters File level, where you can define or redefine any of the attributes in the file. (*First file info*rmation, *then attributes.*)
>
> **ATTRIBUTES-ONLY** provides a shortcut to File level without letting you edit the file definition form. As you use Define mode, you'll find you often want to go to File level to define or redefine an attribute, but you don't want to change the file name or description. For this reason, ATTRIBUTES-ONLY is the default choice.

☒ If you press ⟨ENTER⟩ after selecting REDEFINE-FILE, without selecting a file, Cornerstone will select the highlighted file from the display, and will assume you want to redefine ATTRIBUTES-ONLY. This method requires the fewest keystrokes (if the file you want to redefine is already highlighted).



Attributes

Whether you use DEFINE-NEW-FILE or REDEFINE-FILE, one result is the same: you end up at File level. Now the menu contains commands that work with attributes in the file. The display shows a list of these attributes, together with some information about each one, such as its type.

Many of the commands are similar to ones at Top level, with "attribute" substituted for "file." Two of the commands are DEFINE-NEW-ATTRIBUTE and REDEFINE-ATTRIBUTE—you can probably guess what they do.

Defining a New Attribute

To define a new attribute in the current file, issue the DEFINE-NEW-ATTRIBUTE command. This command works similarly to DEFINE-NEW-FILE at Top level: you define the attribute by filling out a form.

Short Form Versus Long Form

DEFINE-NEW-ATTRIBUTE has two options: SHORT-FORM and LONG-FORM.

When you define an attribute, you must supply Cornerstone with certain information about it, such as its name and type. Whereas there are only two characteristics that apply to files (*File name* and *Description*), for attributes there are many. Of these, however, only three are essential: the attribute's name, description, and type. If you select the SHORT-FORM option, Cornerstone prompts you to supply only these three attribute characteristics. LONG-FORM lets you specify all the attribute characteristics. If you want to define your database quickly, without worrying about every detail, use the short form. To make the process of defining a database easier for beginning users, SHORT-FORM is the default option.

☒ Despite the preceding remarks, you are encouraged to use the long form, since it contains many useful characteristics. For example, in the long form for numbers you can tell Cornerstone to display dollar signs or commas.

The Short Form

Attribute Name. You must give every attribute a name, and as with files, Cornerstone will advise you if you attempt to use an illegal name. (For example, you can't use a name that's already the name of a file.)

Description. The description is used just as for files. You needn't supply a description, but if you do, Cornerstone will use it in help messages that relate to the attribute. The attribute description is intended to serve as a helpful reminder both to you and to anyone else who uses your database. You can type whatever you want for the description, such as "Gross yearly earnings, in quadrillions of dollars."

Type. When you define an attribute, you must tell Cornerstone what type of value the attribute can have. You must enter one of the following types. (You can display this list in Edit mode by pressing ⟨OPTIONS⟩ when the cursor is in the *Type* field.)

□ String

□ Number

□ Integer

□ Date

□ Time

□ Boolean

□ Enumerated

□ Derived

□ Subrecord

These types are described in more detail in the preceding chapter.

☒ You cannot change the type of an attribute that contains data in most cases. See the restrictions listed at the end of the section, Changing an Attribute.

While *Attribute Name, Description,* and *Type* are the three basic pieces of information that define an attribute, Cornerstone sometimes requires more information. When you define a derived attribute, Cornerstone needs to know the derivation expression, which determines how the value is derived. When you define an enumerated attribute, Cornerstone asks you to supply enumerated values for the attribute. In these cases, the short form contains fields for you to enter the required information.

The Long Form

When you select the LONG-FORM option to DEFINE-NEW-FILE, Cornerstone presents you with the usual short form to fill out; however, when you finish editing the form, it expands, prompting you to supply more information about the attribute. These pieces of information, or *attribute characteristics,* are described below. The characteristics that apply for an attribute depend on the attribute's type, so the long form is different for each type of attribute. Below is a sample long form for a number attribute.

```
Attribute name: PRICE
Description: Price per item.
Type: NUMBER
Default display width: 8
Number of post-decimal digits: 2
Display commas?: No
Display dollar sign?: No
Display negative numbers using parentheses?: No
Display zero for numbers less than one?: Yes (e.g., 0.123)
Use scientific notation?: Never
Minimum number value: 0.00
Maximum number value: 999.99
Maximum number of values: 1
Unique values?: No
Indexed?: No
Mandatory?: Yes
Changeable?: Yes
Restrict values to file:
Initial value:
Maintained?: No
```

Characteristics of Attributes

This section contains a list of all the attribute characteristics that can appear in the long attribute definition form. You may wish to merely skim it now, or even skip it entirely and go directly to the next section, Changing an Attribute.

Constraints

The following characteristics are *constraints.* The constraints are checked whenever you *create or change* a record in Update mode. You can change the following constraints anytime in Define mode. If you specify or change any constraints *after* you have already entered data in your file, the existing data is *not* checked against the constraint. The constraint applies only to data that you subsequently enter.

If you specify a constraint for an attribute that has an initial value expression, then when you create or change a record in Update mode, the initial value will not be filled in if it violates any of the constraints. Instead, Cornerstone will leave the value empty.

Maximum length is used for string attributes only. You can use *Maximum length* to restrict the length of values that can be entered in your file. For example, you might set *Maximum length* for *Zip* to be 5 or 10.

Maximum number of values lets you specify the number of values an attribute can have. If you enter 1, the attribute is single-valued. Otherwise the attribute can have as many values as you specify, up to 255. You make an attribute multi-valued by entering a value greater than one.

☒ Once you enter data in your file, you may not change a single-valued attribute to be multi-valued, or vice versa.

Unique values? If you enter "Yes," Cornerstone will compare the value you enter with all other records in the file, and will not permit you to enter a value that's already used. For example, the *Name* in the *Customer* file is constrained to have unique values. This constraint is discussed in chapter 9, "Planning Your Database."

Mandatory? If you enter "Yes," Cornerstone will *require* that you enter a value for this attribute any time you create a new record in the file. This constraint is discussed in chapter 9, "Planning Your Database."

Changeable? tells Cornerstone whether or not to allow the value to be changed once a record is created. If you enter "No," then once you have created a value for the attribute and left Update mode, you may not then ever change the value. This can be used to prevent yourself and others from accidentally (or intentionally) changing data that is critical.

Restrict values to file. This characteristic lets you specify a *cross-file range constraint.* (Cross-file range constraints are explained in chapter 9, "Planning Your Database.") You must enter an expression of the form *file.attribute*, where *attribute* is the name of the attribute in the file named *file* that you want to restrict values to. The file may or may not be related to the current file.

Minimum value and **Maximum value**. The *Minimum value* and *Maximum value* constraints apply to numbers, integers, dates, and times. You can enter any value of the appropriate type, and Cornerstone will check that the value you enter is not less than (or greater than) the constraint. For example, if you enter "0.00" for *Minimum number value,* Cornerstone will prevent you from entering a negative number.

Display Information

The following characteristics determine how the values for an attribute are displayed and printed. They have no effect on the value itself.

Display width. The number you enter in this field determines the width of the field for the attribute whenever you issue the COLUMN or DETAILED commands. This is only the *default* width. You can change the width for a particular form in Format mode.

Number of post decimal digits. This characteristic specifies the number of digits to display following the decimal point for a number attribute. For example, if you enter "4," then 1.2345678 displays as 1.2346 (numbers are rounded appropriately).

Display commas? If you enter "Yes," Cornerstone displays commas every three digits. For example, 123456789 displays as 123,456,789. This characteristic applies to numbers and integers.

Display dollar sign? If you enter "Yes," Cornerstone displays a dollar sign to the left of the value. For example, 12345.67 displays as $ 12345.67. The dollar sign is displayed at the left edge of the field, and the number is right-justified within the field. This characteristic applies to numbers and integers.

Display negative numbers using parentheses? If you want negative numbers to display in parentheses, such as in a financial application, enter "Yes" for this field. This characteristic applies to numbers and integers.

Display zero for numbers less than one? Normally, a number such as 0.123 displays as 0.123, but if you want to suppress the leading zero, so it displays as .123, enter "No" here. Suppressing the zero is useful for numbers that represent a percentage, such as a baseball player's batting average. This characteristic applies only to numbers.

Use scientific notation? You can enter one of three values in this field: "Always," "Never," or "When necessary." The value you enter determines when Cornerstone displays the value using scientific notation. Scientific notation is used to represent very large and very small numbers. For example, 1000000000 is represented as 1e9 (a one followed by 9 zeroes). This characteristic applies to numbers and integers.

The following characteristics apply to *dates*.

Display style for date. Cornerstone allows you to select from a generous number of date display styles. To see all the styles that are available, press the ⟨OPTIONS⟩ key when the cursor lies in this field. The choices are lettered A, B, C, and so on, so all you have to do to select the style you want is enter the letter.

Display year? This characteristic tells Cornerstone whether, and how, to display the year in a date. If you enter "No," the year is not displayed. If you enter "Short," the year is displayed using two digits (for example, 84). If you enter "Long," the year is displayed using four digits (for example, 1984).

Display day of week? This characteristic tells Cornerstone whether, and how, to display the day of week in a date. If you enter "No," the day of week is not displayed. If you enter "Short," the day is displayed using an abbreviation (for example, Wed). If you enter "Long," the full day is displayed (for example, Wednesday).

Display day and month in uppercase? This characteristics determines whether or not the day and month are displayed in uppercase. If you answer "No," only the first letters of the day and month are capitalized.

The following characteristics apply to *times*.

Display style for time. You can select from a number of display styles for times. For example, you can display times in twelve or twenty-four hour notation. To see the possible choices, press the ⟨OPTIONS⟩ key when the cursor lies in this field.

Display AM/PM in uppercase? This characteristic determines whether or not the AM/PM indication is displayed in upper or lower case. Of course, its value is relevant only when you've chosen a display style that shows AM/PM.

Miscellaneous Characteristics

Derivation. This field is used to specify the derivation expression for a derived attribute. To see what you can type, press the ⟨OPTIONS⟩ key. The chapter "Entering Information into Forms" in *Owner's Handbook II* lists the kinds of expressions you can enter.

Initial value. This field is used to specify an initial value expression for a (primitive) attribute. To see what you can type, press the ⟨OPTIONS⟩ key. The chapter "Entering Information into Forms" in *Owner's Handbook II* lists the kinds of expressions you can enter. If you specify an initial value, whenever you create a new record in the file, Cornerstone will compute the initial value for the attribute and fill it in. You can then change the value if you like. Initial values are discussed in "Planning Your Database," and also in "The Recompute Utility" and "Efficiency With Cornerstone " in *Owner's Handbook II*.

Enumerated values. This field must be filled in for attributes you define to be *enumerated*. Enter the values you want the attribute to be able to have (such as Red, Green, Blue). This is a multi-valued field; you can enter up to 255 different values. Once you've entered data for the attribute, you may not delete or insert values, but you may change existing ones, and you may append new values to the end of the list.

Indexed? If you enter "Yes," Cornerstone will create an index for the attribute. An index allows the attribute to be accessed more quickly, and speeds up certain operations, such as sorting and selection. The chapter "Efficiency With Cornerstone" in *Owner's Handbook II* discusses when to make an attribute indexed. String, number, integer, date, time, and enumerated attributes can be indexed.

Maintained? This characteristic tells Cornerstone whether or not the attribute is one of the *maintained* attributes that are used with the RECOMPUTE command. Maintained attributes are discussed in

"The Recompute Utility" and "Efficiency With Cornerstone" in *Owner's Handbook II*. Any primitive attribute can be maintained, but it only makes sense for an attribute to be maintained if it also has an initial value expression.

Changing an Attribute

Once you've defined an attribute, you may wish to change it—or, if you used the short form to create the attribute, you might want to go back and enter some of the characteristics in the long form. REDEFINE-ATTRIBUTE is the command to use in either of these cases.

To change an attribute, select REDEFINE-ATTRIBUTE from the menu and then select the attribute you want to redefine.

Cornerstone presents you with the long form, which you can edit to change the attribute's characteristics. To change a characteristic, simply move to the appropriate field and enter the new value.

```
Attribute name: TOTAL_PURCHASES
Description: The total of the purchases, excluding the Gadsden
             Purchase.
Type: NUMBER
Default display width: 12
Number of post-decimal digits: 2
Display commas?: Yes
Display dollar sign?: Yes
Display negative numbers using parentheses?: No
Display zero for numbers less than one?: Yes (e.g., 0.123)
Use scientific notation?: When necessary
Minimum number value:
Maximum number value:
Maximum number of values: 1
Unique values?: No
Indexed?: No
```

```
Mandatory?: No
Changeable?: No
Restrict values to file:
Initial value: SUM (ORDER.TOTAL_PRICE)
Maintained?: Yes
```

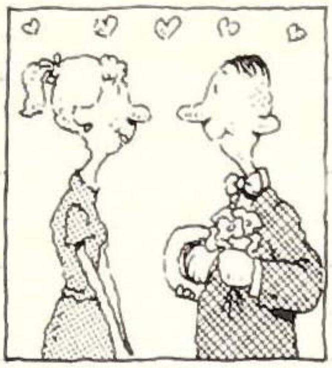
☒ If you press ⟨ENTER⟩ after selecting REDEFINE-ATTRIBUTE, without selecting an attribute, Cornerstone will select the attribute highlighted in the display.

There are a few restrictions regarding what you may enter in the long form. You needn't be too concerned about them—Cornerstone will inform you whenever you attempt something illegal. You can then press the ⟨HELP⟩ key to find out what to do. For those who are curious, the restrictions are described below.

□ You may not make a single-valued attribute multi-valued or vice versa, once the file contains data for the attribute. You can change the *Maximum number of values* of a multi-valued attribute to one, but this does not make it single-valued. (The restrictions imposed on multi-valued attributes still apply.)

□ You may not change the *Type* of an attribute when the file contains data for the attribute, or when the attribute affects other attributes in your database (that is, when other attributes reference the attribute in derivation or initial value expressions). You may, however, make a derived attribute primitive (or vice versa)—even when there's data—as long as you preserve the data type. For example, you can change a Number to be Derived, as long as the derivation you supply results in a number. You can change freely between Number and Integer, even when the file contains data.

☒ There are sometimes ways to change the type of an attribute that contains data, and to make a single-valued attribute multi-valued or vice versa. "The Recompute Utility" in *Owner's Handbook II* explains this.



Relationships

One of the most powerful features of Cornerstone is that it lets you define relationships which allow information to be shared among the files in your database. The preceding chapter illustrates some of the benefits of relationships. Now you'll learn how to define them.

How to Start a Relationship

You define a relationship between two files using the DEFINE-RELATIONSHIP command at File level. This command requires you to select three things.

The other file you want to define the relationship with. Cornerstone allows you to select any file or subfile that isn't already related to the current file. This file is called the *related* file (even though the relationship goes both ways). You can only define one relationship between a pair of files, and you cannot define a relationship between a file and itself.

The attribute in the current file. This is the attribute in the current file you want to relate with an attribute in the related file. You can select any attribute that appears in the menu. You cannot define a relationship for an attribute of type Enumerated, Boolean, or Subrecord, and you cannot define a relationship for an attribute that is multi-valued.

The attribute in the related file. Again, you can select any attribute that appears in the menu. The menu contains only attributes in the related file that are of the same type as the attribute you chose from the current file.

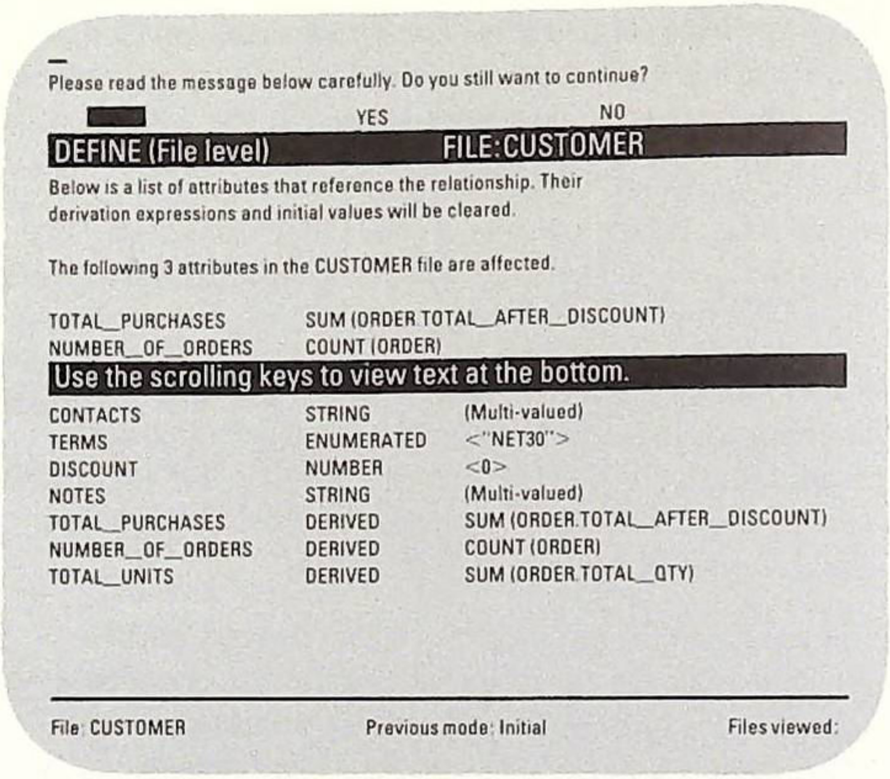
Once you have defined a relationship between two files, you can share information between them by defining derived attributes and initial value expressions.

Breaking Up Isn't Hard to Do

While relationships are often a good thing, sometimes they don't work out. Fortunately, when this happens, you can use the REMOVE-RELATIONSHIP command.

When you select REMOVE-RELATIONSHIP, Cornerstone prompts you to select the related file you want to end the relationship with. The menu shows only files that are related to the current file. You don't have to specify the attributes, since Cornerstone knows which they are.

Removing a relationship between two files may affect attributes that use the relationship, either in derivation or initial value expressions. Cornerstone will explain these effects and ask you for confirmation before actually removing the relationship. The figure below shows what would happen if you blithely attempted to remove the relationship between the *Customer* and *Order* files in the Sample Database.



```
Please read the message below carefully. Do you still want to continue?
                         YES                              NO
DEFINE (File level)                      FILE:CUSTOMER
Below is a list of attributes that reference the relationship. Their
derivation expressions and initial values will be cleared.

The following 3 attributes in the CUSTOMER file are affected.

TOTAL_PURCHASES      SUM (ORDER.TOTAL_AFTER_DISCOUNT)
NUMBER_OF_ORDERS     COUNT (ORDER)
Use the scrolling keys to view text at the bottom.
CONTACTS             STRING        (Multi-valued)
TERMS                ENUMERATED    <"NET30">
DISCOUNT             NUMBER        <0>
NOTES                STRING        (Multi-valued)
TOTAL_PURCHASES      DERIVED       SUM (ORDER.TOTAL_AFTER_DISCOUNT)
NUMBER_OF_ORDERS     DERIVED       COUNT (ORDER)
TOTAL_UNITS          DERIVED       SUM (ORDER.TOTAL_QTY)

File: CUSTOMER               Previous mode: Initial               Files viewed:
```

Relationships are the bonds that tie together information contained in separate files. If you remove a relationship between two files, the bond is severed, and the two files can no longer share information.

Defining Subrecords

To define a subrecord attribute, simply enter "Subrecord" as the *type* when you define the attribute. The only attribute characteristic that applies to subrecord attributes is *Maximum number of values*. In most applications, you will want the subrecord to be multi-valued. For this reason, the maximum number of values defaults to the maximum number Cornerstone allows (which is 255). You needn't decrease this value in an attempt to save disk space, since Cornerstone uses only as much space as is required to store values you've entered.

When you're done editing the attribute definition form for a subrecord attribute, and press ⟨DONE⟩, Cornerstone takes you to Subfile level of Define mode. In fact, this is how you get to Subfile level: either by defining or redefining a subrecord attribute. Subfile level is completely analogous to File level. The commands work on *sub*attributes in the current *sub*file instead of attributes in the current file. Since these commands operate identically to their File level counterparts, we list them only for reference.

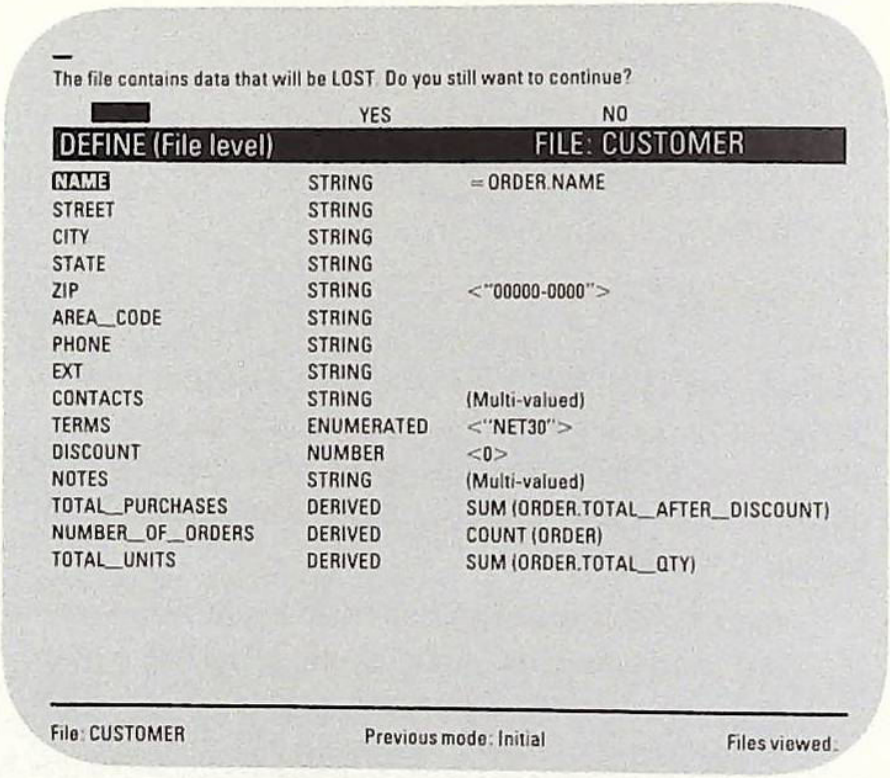
- □ DEFINE-NEW-SUBATTRIBUTE
- □ REDEFINE-SUBATTRIBUTE
- □ DELETE-SUBATTRIBUTE
- □ DEFINE-RELATIONSHIP
- □ REMOVE-RELATIONSHIP

Deleting Files and Attributes

At some point you may decide that you no longer need a particular file or attribute. This is especially likely in the early stages of database design, when you experiment with different approaches. There's a DELETE command for each level in Define mode.

> **DELETE-FILE** is available at Top level and allows you to delete an entire file, including any data it may contain. You must remove all relationships defined for the file before you can delete it. If you attempt to delete a file that contains data, Cornerstone will request confirmation before deleting it.
>
> **DELETE-ATTRIBUTE** is available at File level and allows you to remove an attribute from a file. You must remove all relationships defined for the attribute before you can delete it. If you attempt to delete an attribute that contains data, Cornerstone will request

confirmation before deleting the attribute, just as with DELETE-FILE. When you attempt to delete an attribute that is used by one or more other attributes in a derivation or initial value expression, Cornerstone displays a message indicating which other attributes are affected, and requests confirmation before deleting the attribute. The figure below shows an example of this.



```
The file contains data that will be LOST. Do you still want to continue?
                          YES                        NO
DEFINE (File level)                           FILE: CUSTOMER
NAME                STRING       = ORDER.NAME
STREET              STRING
CITY                STRING
STATE               STRING
ZIP                 STRING       <"00000-0000">
AREA_CODE           STRING
PHONE               STRING
EXT                 STRING
CONTACTS            STRING       (Multi-valued)
TERMS               ENUMERATED   <"NET30">
DISCOUNT            NUMBER       <0>
NOTES               STRING       (Multi-valued)
TOTAL_PURCHASES     DERIVED      SUM (ORDER.TOTAL_AFTER_DISCOUNT)
NUMBER_OF_ORDERS    DERIVED      COUNT (ORDER)
TOTAL_UNITS         DERIVED      SUM (ORDER.TOTAL_QTY)

File: CUSTOMER            Previous mode: Initial            Files viewed:
```

DELETE-SUBATTRIBUTE is used to delete a subattribute from a subfile, and is available at Subfile level. This command works analogously to DELETE-ATTRIBUTE.

Printing Your Database Definition

As you are designing your database, you may want to get a description of it on paper, so you can ponder your design. Or you may just want to get a permanent record of a design that you're satisfied with. The PRINT command in Define mode lets you print your database definition. There's a print command at each level of Define mode.

> At **Top level** you can print the definition of a single file or all files in your database.
>
> At **File level** you can print the definition of a single attribute or all attributes in the current file.
>
> At **Subfile level** you can print the definition of a single (sub)attribute or all (sub)attributes in the current subfile.

With each of these commands, Cornerstone gives you the option of printing to the printer or to a disk file. To see what the printouts look like, try printing them!

Leaving Define Mode

To leave one level of Define mode and return to the previous one, press the ⟨DONE⟩ key. If you press ⟨DONE⟩ when you're at Top level, Cornerstone will leave Define mode and return to Initial mode.

If you want to leave Define mode and return directly to Initial mode from *any* level, just issue the ALL-DONE command.

When leaving Define mode, it sometimes takes a little time while Cornerstone updates your database to reflect the changes you have made. Sometimes, for example, Cornerstone must create indexes for attributes in one or more files. When this happens, Cornerstone will let you know. Please be patient and allow the operation to complete.

That's a Wrap

This and the preceding chapter contain all the information you need to know to design and build any database using Cornerstone. Once you've read these chapters, you're ready to put on your designer's cap and experiment with different database designs using Cornerstone. If you like, you can use either the Sample Database or the Client Tracking system included with Cornerstone as a starting point for your own database.

Finally, at the risk of being repetitive, we take this opportunity once again to emphasize the importance of a good design. The more effort you put into your database design, the more easily you will be able to retrieve the information you want, when you want it. Good luck!

CUSTOMIZATION AND INITIAL MODE

11 CHAPTER

CHAPTER 11

CONTENTS

Customization and Initial Mode

Introduction

A Cornerstone session begins and ends in Initial mode. In Initial mode you can issue commands that take you to Define mode, where you set up the structure of your database; to Update mode, where you enter data to create records; and to View mode, where you look at your records. You can read about those modes elsewhere in *Owner's Handbook I*. This chapter explains how to use Initial mode's EXIT, CUSTOMIZE, and DELETE commands, and gives an overview of the utilities.

The EXIT Command

You wouldn't leave the tenth floor of a building by jumping out of a window, would you? Sure, it's the fastest way down, but you're likely to break something when you reach the ground. It's much safer to walk to the elevator, take the elevator down, and walk out the front door. Similarly, the only way to end a Cornerstone session without risk of ruining your data is to issue the EXIT command in Initial mode. This command takes a few seconds to put the database in an orderly state, then returns you to your operating system. (If you mistakenly begin an operation that takes a long time to complete, you can press Ctrl-Break to safely stop the operation and exit Cornerstone.) Do not try to exit the system in any other way, such as turning off the power to your computer, removing data diskettes, or rebooting the system.

The CUSTOMIZE Command

Cornerstone allows you to tailor certain aspects of the system to your liking. When you issue the CUSTOMIZE command, Cornerstone displays a form whose fields refer to properties of the system you can adjust. You edit the form as you edit any other form in Cornerstone: just use the editing keys, and enter your value. When you press ⟨DONE⟩, the values take effect. Cornerstone remembers the values between sessions, so you don't have to re-enter them every time you start a new session.

Cornerstone provides a value for each of these fields; if you are satisfied with a particular value, leave it as it is. Here are the properties of the system you can adjust with the CUSTOMIZE command:

Edit mode type-in style. Cornerstone provides the value "Overstrike," meaning any character you type in Edit mode will replace the character the cursor is on. If you change this value to "Insert," any character you type in Edit mode will be inserted before the character the cursor lies on. You can always override the effect of this entry by pressing ⟨INS/OVS⟩ in Edit mode.

Format mode type-in style. As in the previous field, Cornerstone provides the value "Overstrike." You can change this value to "Insert." As in Edit mode, you can always override the effect of this entry by pressing ⟨INS/OVS⟩.

Menu size. This value tells Cornerstone how many menu lines to display. The default value is 2. You can enter an integer from 0 to 8. When you become more familiar with the system, you may want to enter a value of zero, leaving more room to display data. Of course, you can display the entire menu at any time by pressing ⟨OPTIONS⟩.

Page overlap. Your entry here tells Cornerstone how many lines to overlap when you press ⟨PAGE-FORWARD⟩ and ⟨PAGE-BACKWARD⟩. The default value is 2; you can enter any integer from 0 to 24.

Horizontal scroll amount. This value tells Cornerstone how many columns to move the window at a time when you press ⟨WINDOW-RIGHT⟩ and ⟨WINDOW-LEFT⟩. The default value is 17; you can enter any integer from 1 to 80.

Default string width. One of the things you can specify when you define an string attribute is the width of the field that displays the attribute. If you don't specify a value when you define the attribute, Cornerstone automatically assigns the value entered here. The default is 60; you can enter any value from 1 to 255.

Display status history? While using the system you may find it helpful to display the status lines from all earlier levels of interaction. They tell you how you got where you are, and which mode you will return to when you press ⟨DONE⟩. A value of "Yes" here displays the status history; the default value "No" displays just the current status line above the data area and a second status line below the data area, showing current file, previous mode, and number of files viewed.

Maximum form width. The value you enter here determines the maximum width of a form. If you enter 100 here, for example, Cornerstone will prevent you from creating a form wider than 100 characters. The default is 255; you can enter a number from 80 (which is the screen width) to 255.

Screen color. Enter "Color" in this field to see Cornerstone in color on your color monitor. The default value is "Monochrome." ("Reinstallation and Saving Space" in *Owner's Handbook II* tells you how to change the colors of the Cornerstone display.)

Ring bell on error? Enter "Yes" in this field if you want to Cornerstone to sound a bell when it displays an error message. The default value is "No."

The DELETE Command

If you want to tidy up your database, the DELETE command in Initial mode lets you delete formats, selection criteria, and logical print devices from your database. (You delete files and attributes, and remove relationships, in Define mode; you delete records in Update mode.)

☒ When you delete a format or selection criteria, you free up disk space *only* for new formats and selection criteria; you do *not* free up disk space for data, attributes, files, or devices. Similarly, when you delete a print device, you free up disk space *only* for new print devices.

To delete a format, issue the DELETE command with the option FORMAT, the name of the file that contains the format, and finally the name of the format. For example, the command

```
DELETE FORMAT CUSTOMER PHONE_BOOK
```

deletes the format called *Phone_Book* from the *Customer* file. You issue the same kind of command to delete selection criteria. For example, the command

```
DELETE SELECTION-CRITERIA CLIENTS EAST_COAST
```

deletes the selection criteria you saved under the name *East_Coast* from the *Clients* file.

To delete a logical print device from the database, issue the DELETE command with the DEVICE option and the name of the device. For example, the command

```
DELETE DEVICE MY_PRINTER
```

deletes the logical print device named *My_Printer* from the database.

Overview of the Utilities

Utilities let you take care of housekeeping tasks for one or more files at a time. The Cornerstone utilities let you back up your data, convert data to and from other computers and software products, and do bulk updates of certain attributes in a file. You use the Cornerstone utilities by issuing the UTILITIES command in Initial mode.

> The **UTILITIES CONVERT** command brings you to Convert mode, where you can convert data entered in other software products for use in Cornerstone, and vice versa. Among the products you can convert from are dBASE II®, PFS®, 1-2-3™, and products using DIF, MailMerge®, and a few other data formats. See "The Convert Utility" in *Owner's Handbook II*.
>
> The **UTILITIES BACKUP/RESTORE** command brings you to Backup mode, where you can back up one or more Cornerstone files or restore files you have backed up. You should back up your Cornerstone files often, because if there is a power failure while you are using Cornerstone, or if someone exits the system improperly, your data might be ruined. You can also use Backup mode to reorganize certain aspects of your database. See "The Backup Utility" in *Owner's Handbook II* to learn how to use the Backup utility.

The UTILITIES RECOMPUTE command lets you perform bulk updates of attributes that have initial values. This utility can be useful if you have added attributes to a file, if you want to change an attribute's type, or if you are using initial values (for their speed) in place of derived attributes. See "The Recompute Utility" in *Owner's Handbook II* to learn how to use the RECOMPUTE command.

INDEX

INFOCOM™

Part No. 10021